Universidade de Aveiro Departamento de Economia, Gestão e Engenharia Industrial
2011

Rui Jorge Ferreira Soares Borges Lopes

Problemas de Localização-Distribuição de Serviços Semiobnóxios: Aproximações e Apoio à Decisão

Location-Routing Problems of Semi-Obnoxious Facilities: Approaches and Decision Support

Universidade de Aveiro Departamento de Economia, Gestão e Engenharia Industrial
2011

Rui Jorge Ferreira Soares Borges Lopes

Problemas de Localização-Distribuição de Serviços Semiobnóxios: Aproximações e Apoio à Decisão

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Gestão Industrial, realizada sob a orientação científica do Doutor Carlos Manuel dos Santos Ferreira, Professor Associado com Agregação do Departamento de Economia, Gestão e Engenharia Industrial da Universidade de Aveiro e da Doutora Maria Beatriz Alves de Sousa Santos, Professora Associada com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Universidade de Aveiro Departamento de Economia, Gestão e Engenharia Industrial
2011

Rui Jorge Ferreira Soares Borges Lopes

Location-Routing Problems of Semi-Obnoxious Facilities: Approaches and Decision Support

Thesis submitted to the University of Aveiro to fulfil the requirements for the degree of Doctor of Philosophy in Industrial Management, made under the scientific supervision of Doctor Carlos Manuel dos Santos Ferreira, Associate Professor with Habilitation at the Department of Economics, Management and Industrial Engineering of the University of Aveiro and Doctor Maria Beatriz Alves de Sousa Santos, Associate Professor with Habilitation in the Department of Electronics, Telecommunication and Informatics of the University of Aveiro.

Aos meus pais

To my parents

o júri

presidente

Doutor Fernando Manuel Bico Marques
Professor Catedrático da Universidade de Aveiro

Doutor Carlos Alberto Henggeler de Carvalho Antunes
Professor Catedrático da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Doutor Joaquim José Borges Gouveia
Professor Catedrático da Universidade de Aveiro

Doutor Jorge Manuel Pinho de Sousa
Professor Associado com Agregação da Faculdade de Engenharia da Universidade do Porto

Doutor Carlos Manuel dos Santos Ferreira
Professor Associado com Agregação da Universidade de Aveiro

Doutora Maria Beatriz Alves de Sousa Santos
Professora Associada com Agregação da Universidade de Aveiro

Doutor Gábor Nagy
Senior Lecturer da *Kent Business School* da Universidade de Kent

Doutor Francisco Alexandre Saldanha da Gama Nunes da Conceição
Professor Auxiliar da Faculdade de Ciências da Universidade de Lisboa

Doutor António Ismael de Freitas Vaz
Professor Auxiliar da Escola de Engenharia da Universidade do Minho

the jury

chairman

Doctor Fernando Manuel Bico Marques
Professor at the University of Aveiro

Doctor Carlos Alberto Henggeler de Carvalho Antunes
Professor at the Faculty of Sciences and Technology of the University of Coimbra

Doctor Joaquim José Borges Gouveia
Professor at the University of Aveiro

Doctor Jorge Manuel Pinho de Sousa
Associate Professor with Habilitation at the Faculty of Engineering of the University of Porto

Doctor Carlos Manuel dos Santos Ferreira
Associate Professor with Habilitation at the University of Aveiro

Doctor Maria Beatriz Alves de Sousa Santos
Associate Professor with Habilitation at the University of Aveiro

Doctor Gábor Nagy
Senior Lecturer at the Kent Business School of the University of Kent

Doctor Francisco Alexandre Saldanha da Gama Nunes da Conceição
Assistant Professor at the Faculty of Sciences of the University of Lisbon

Doctor António Ismael de Freitas Vaz
Assistant Professor at the School of Engineering of the University of Minho

agradecimentos

Este trabalho não seria possível sem as valiosas sugestões dos Professores Carlos Ferreira e Beatriz Sousa Santos, respectivamente meu orientador e co-orientadora. A ambos agradeço o empenho, interesse e valiosas sugestões que permitiram o correcto encaminhamento e conclusão deste trabalho.

Estou igualmente grato ao Professor Frank Plastria pela recepção, supervisão e apoio proporcionados durante a minha estadia em Bruxelas.

Ao Departamento de Economia, Gestão e Engenharia Industrial, na pessoa da Professora Helena Alvelos, presidente do departamento, pelas condições favoráveis proporcionadas durante o decorrer deste trabalho.

Ao Departamento MOSI da *Vrije Universiteit Brussel*, na pessoa do Professor Marc Despontin, presidente do departamento, pelas facilidades proporcionadas durante o período que passei nesse departamento.

Ao Centro de Investigação Operacional, da Universidade de Lisboa pelo apoio financeiro prestado ao longo do trabalho que me permitiu apresentá-lo em diversas reuniões científicas.

À Fundação para a Ciência e a Tecnologia pelo apoio financeiro dado sob a forma de bolsa de doutoramento.

A todos os meus colegas e amigos da Universidade de Aveiro.

Por fim, aos meus pais e irmãos pelo apoio, auxílio e motivação que me prestaram durante a elaboração deste trabalho.

acknowledgements

This work could not have been possible without the valuable suggestions of Professors Carlos Ferreira and Beatriz Sousa Santos, respectively my supervisor and co-supervisor. To both my appreciation for the commitment, interest and valuable suggestions which allowed the correct guidance and completion of this work.

I am also grateful to Professor Frank Plastria for the warm welcome and supervision provided during my stay in Brussels.

To the Department of Economics, Management and Industrial Engineering, in the person of Professor Helena Alvelos, head of the department, for the favourable conditions provided during the course of this work.

To the Department MOSI of the *Vrije Universiteit Brussel*, in the person of Professor Marc Despontin, head of the department, for the facilities provided during the time spent in that department.

To the Operations Research Center of the University of Lisbon for the financial support provided throughout this work which allowed me to present it at various scientific meetings.

To the Portuguese National Science Foundation for the financial support in the form of a research grant.

To all my colleagues and friends of the University of Aveiro.

Finally, to my parents and brothers for the support, help, and motivation provided during the course of this work.

palavras-chave

Logística, localização-distribuição, multi-objectivo, serviços semiobnóxios, sistemas de apoio à decisão.

resumo

A presente tese resulta de um trabalho de investigação cujo objectivo se centrou no problema de localização-distribuição (PLD) que pretende abordar, de forma integrada, duas actividades logísticas intimamente relacionadas: a localização de equipamentos e a distribuição de produtos.

O PLD, nomeadamente a sua modelação matemática, tem sido estudado na literatura, dando origem a diversas aproximações que resultam de diferentes cenários reais. Importa portanto agrupar as diferentes variantes por forma a facilitar e potenciar a sua investigação. Após fazer uma revisão e propor uma taxonomia dos modelos de localização-distribuição, este trabalho foca-se na resolução de alguns modelos considerados como mais representativos. É feita assim a análise de dois dos PLDs mais básicos (os problema capacitados com procura nos nós e nos arcos), sendo apresentadas, para ambos, propostas de resolução. Posteriormente, é abordada a localização-distribuição de serviços semiobnóxios. Este tipo de serviços, ainda que seja necessário e indispensável para o público em geral, dada a sua natureza, exerce um efeito desagradável sobre as comunidades contíguas. Assim, aos critérios tipicamente utilizados na tomada de decisão sobre a localização destes serviços (habitualmente a minimização de custo) é necessário adicionar preocupações que reflectem a manutenção da qualidade de vida das regiões que sofrem o impacto do resultado da referida decisão.

A abordagem da localização-distribuição de serviços semiobnóxios requer portanto uma análise multi-objectivo. Esta análise pode ser feita com recurso a dois métodos distintos: não interactivos e interactivos. Ambos são abordados nesta tese, com novas propostas, sendo o método interactivo proposto aplicável a outros problemas de programação inteira mista multi-objectivo.

Por último, é desenvolvida uma ferramenta de apoio à decisão para os problemas abordados nesta tese, sendo apresentada a metodologia adoptada e as suas principais funcionalidades. A ferramenta desenvolvida tem grandes preocupações com a interface de utilizador, visto ser direccionada para decisores que tipicamente não têm conhecimentos sobre os modelos matemáticos subjacentes a este tipo de problemas.

keywords

Logistics, location-routing, multi-objective, semi-obnoxious facilities, decision support systems.

abstract

This thesis main objective is to address the location-routing problem (LRP) which intends to tackle, using an integrated approach, two highly related logistics activities: the location of facilities and the distribution of materials.

The LRP, namely its mathematical formulation, has been studied in the literature, and several approaches have emerged, corresponding to different real-world scenarios. Therefore, it is important to identify and group the different LRP variants, in order to segment current research and foster future studies. After presenting a review and a taxonomy of location-routing models, the following research focuses on solving some of its variants. Thus, a study of two of the most basic LRPs (capacitated problems with demand either on the nodes or on the arcs) is performed, and new approaches are presented. Afterwards, the location-routing of semi-obnoxious facilities is addressed. These are facilities that, although providing useful and indispensible services, given their nature, bring about an undesirable effect to adjacent communities. Consequently, to the usual objectives when considering their location (cost minimization), new ones must be added that are able to reflect concerns regarding the quality of life of the communities impacted by the outcome of these decisions.

The location-routing of semi-obnoxious facilities therefore requires to be analysed using multi-objective approaches, which can be of two types: non-interactive or interactive. Both are discussed and new methods proposed in this thesis; the proposed interactive method is suitable to other multi-objective mixed integer programming problems.

Finally, a newly developed decision-support tool to address the LRP is presented (being the adopted methodology discussed, and its main functionalities shown). This tool has great concerns regarding the user interface, as it is directed at decision makers who typically don’t have specific knowledge of the underlying models of this type of problems.

Contents

List of Figures

List of Tables

List of Acronyms

AGS	Active guided search
BSSS	Big square small square
CARP	Capacitated arc routing problem
CLRP	Capacitated location-routing problem
CVRP	Capacitated vehicle routing problem
DM	Decision maker
DSS	Decision support system
DST	Decision-support tool
EAM	Extended augment-merge
EM	Extended merge
GLS	Guided local search
GRASP	Greedy randomized adaptive search procedure
GUI	Graphical user interface
HAZMAT	Hazardous material
HCI	Human-computer interaction
LARP	Location-arc routing problem
LRP	Location-routing problem
LS	Local search
MOIP	Multi-objective integer programming
MOMIP	Multi-objective mixed integer programming
NSGA-II	Non-dominated sorting genetic algorithm II
PVN	Penalty variable neighbourhood
RAD	Rapid application development
RCL	Restricted candidate list
RCX	Route copy crossover
RUP	Rational unified process
TS	Tabu search

TSP	Travelling salesman problem
UML	Unified modeling language
VNS	Variable neighbourhood search
VRP	Vehicle routing problem
WMS	Web map service
XAML	Extensible application markup language
XML	Extensible markup language
XP	Extreme programming

Chapter 1

Introduction

Logistics plays an important role within organizations, dealing with the planning and control of material flows and related information. There is always a cost (not necessarily economic) associated with the management, maintenance, manipulation, and storage of materials. Although this cost cannot be eliminated, its optimization may become a strong competitive argument. Therefore, good decisions regarding the design and activities of logistics systems are of the utmost importance to managers.

In this thesis two of the most important decisions in logistics systems activities will be addressed: the location of facilities and the distribution of materials.

When a decision maker (DM) addresses the design and activities of logistics systems, among others, (s)he must determine the location of facilities and the distribution of materials. In order to do so, several aspects have to be considered, some hardly measurable.

One of the aspects, and often the most relevant, is cost. The cost of locating facilities typically plays an important role in the overall cost of logistics systems, as it may not be easily changed and affects the remaining logistics activities. However, if the cost of locating a facility is to be correctly obtained, the distribution of materials must also be considered. These are highly related logistics activities, and thus, should be considered simultaneously.

The role of operations research (a field of applied mathematics) is to provide quantitative tools to DMs, based on which they can support decisions regarding these activities. In operations research, these decisions have already been approached in an integrated fashion, being the underlying problem named location-routing problem (LRP). The LRP has had several variants reflecting the different main real-world scenarios (which have also led to a currently somewhat disperse body of knowledge).

When intending to determine the location of facilities, cost is, in most cases, the most important aspect; however, for some facilities other aspects gain relevance. This is the case with semi-obnoxious facilities. These are facilities that, although providing useful and indispensable services, due to their nature, bring about undesirable effects to communities (e.g. affecting people's quality of life and urban environment, by bringing about traffic chaos, noise, and pollution), when installed in their vicinity. Examples of such facilities are landfills, prisons, fire stations, power plants, airports, and hospitals. The specific characteristics of semi-obnoxious facilities requires that the undesirable effect must also be considered (without completely disregarding cost); as the general public is increasingly gaining environmental awareness and is often not willing to lose quality of life.

The abovementioned situation suggests the use of multi-objective approaches which, unlike single-objective ones (where complete ordering of solutions is possible), requires the contribution of DMs for the ordering and evaluation of solutions (due to the existing trade-off between cost and obnoxious effect, for instance).

The different facility location (and routing) decisions discussed previously are the subject of this thesis, and are studied using operations research methodologies. The underlying single- and multi-objective problems are formally defined, and new approaches proposed and validated. When addressing the multi-objective LRP model, two different types of approaches can be used: non-interactive and interactive methods.

Non-interactive multi-objective methods are more prone to be used in situations where there is no knowledge of the preferences of the DM (or the DM either has unclear preferences or little knowledge of the problem at hand); thus methods attempt to obtain all non-dominated solutions, to later be evaluated. These methods may lead to excessive computation time and the generation of an overwhelming number of solutions, which may be difficult for DMs to analyse; moreover, most of them are not interesting to DMs.

In interactive multi-objective methods, the computation time is decreased as only solutions found to be interesting to DMs are generated. This requires alternation between human intervention (DM) and computation phases. The difficulties encountered in such methods are also shared by other multi-objective mixed integer programming (MOMIP) problems. Therefore, the newly developed method can be used on other such problems.

The main objective of the developed approaches is to support DMs in their location decisions. However, DMs often do not possess specific knowledge of LRP models, making these studies seldom known or used outside the academic community. Moreover, the decision-making process usually requires a lot of experience on either the addressed problem or the approaches to solve it. With computation tools the need for acquired experience, as well as the time to obtain solutions for a specific scenario may decrease significantly. This justifies the importance to develop such a tool: that is able to incorporate advanced algorithms while making easy the visualization (and editing) of data regarding these problems.

A proposal for a decision-support tool (DST) for LRPs is thus presented. As the success of such a tool depends greatly on its correct development and fulfilment of requirements (regarding performance, reliability, robustness, usability, etc.), main software development methodologies in the literature are reviewed, and key success aspects identified. As the proposed DST is mostly directed at users with little knowledge of modelling or optimization, special emphasis should be given to the user interface.

The user interface should aim at allowing the exploration of the solution-finding process in a way easily understandable by the user, facilitating it, and eventually fostering greater insight of the problem at hand. Moreover, by presenting an interface easy to work with and understand, the general public may have easier access and further understanding of the decision process involved in many of the current facility installation decisions.

To sum up, the previously mentioned considerations motivated the development of this thesis, which presents work in three (complementary) dimensions: location-routing, multi-objective, and decision support. The underlying research subject (which requires addressing the three aforementioned dimensions) is therefore if multi-objective approaches, for LRPs concerning semi-obnoxious facilities, can contribute to improve the quality of decisions.

1.1 Objectives

The main objective of this thesis is to develop new approaches that enable to improve decision support in the subject of location-routing. Both single-objective as well as multi-objective problems are addressed, albeit main contributions are regarding the multi-objective case (used when locating semi-obnoxious facilities).

The overall objective can be achieved through the following specific objectives:

- to review and segment current research on LRPs
- to develop new approaches for basic (single-objective) LRPs
- to review location models addressing undesirable (obnoxious and semi-obnoxious) facilities
- to develop non-interactive and interactive multi-objective approaches for the location-routing of semi-obnoxious facilities
- to implement the proposed approaches in a DST directed at DMs
- to ensure the developed DST is easy to learn and work with.

Based on the need to develop new approaches for LRPs, current research is to be reviewed and segmented, in order to identify basic models and current approaches in the literature. With the intent of increasing the knowledge on some of the basic (single-objective) LRPs, new approaches are to be developed and validated.

As the study of the location-routing of semi-obnoxious facilities is another of the main goals of this thesis, models considering only location of undesirable facilities (both obnoxious and semi-obnoxious) are to be reviewed. The goal is to identify main objectives and modelling issues in order to incorporate them when tackling the location-routing of semi-obnoxious facilities. To solve the resulting multi-objective problem, non-interactive or interactive approaches are required to be employed (both corresponding to different decision-making scenarios). Aiming at supporting decisions for most of real-world scenarios, both a non-interactive and an interactive approach are to be developed and analysed.

To effectively improve decision support in these problems, not only new approaches should be developed, but they should also be made available. This can be achieved by developing a DST, directed at DMs (from which the general public may also profit), that is able to incorporate the developed approaches. In order to accomplish this goal, the tool's development must have great concerns regarding the user interface. To that extent, usability tests are to be performed to evaluate the usability of the tool, ensuring it is easy to learn and work with.

1.2 Thesis Outline

This thesis is organized in eight chapters which address several relevant issues regarding location-routing, multi-objective, and decision support.

A brief introduction and general description of the studies made in this thesis is presented in Chapter 1, where some general considerations are made, the underlying objectives defined, and the outline of the thesis described.

In Chapter 2 an overview of facility location and vehicle routing (two of the main activities of logistics systems) is provided, in which some of the problems in the literature addressing them are presented. The integrated location-routing approach is then analysed, where several works with the different LRP variants, objectives, and approaches can be found. This motivated the development of a comprehensive taxonomy which was proposed in the same chapter. The taxonomy separates papers in the literature according to, firstly, the physical characteristics of the models, secondly, the algorithmic approach and the number of objectives.

From the existing LRP variants described in Chapter 2, two of the most basic (single-objective) problems are chosen to be studied in Chapter 3: the capacitated LRP (CLRP) and the location-arc routing problem (LARP). For the CLRP, a formal definition is provided, existing approaches are reviewed, and a new metaheuristic (active guided search – AGS) is proposed. The AGS metaheuristic is tested using three sets of benchmark instances from the literature, proving to be competitive when compared with other approaches. For the LARP, a formal definition is given and, as the literature review revealed this problem as scarcely studied, new constructive methods, improvement heuristics, and metaheuristic approaches are proposed. Moreover, a new set of benchmark instances had to be devised, allowing to compare the different proposals and draw some conclusions.

As one of the purposes of this thesis is to study the location-routing of semi-obnoxious facilities, in Chapter 4, works (and corresponding models) addressing, firstly, solely facility location, then the location-routing of undesirable facilities is reviewed. It is shown that this problem is inherently multi-objective, and the scarcity of multi-objective location-routing models motivates the formal definition and use of a new multi-objective CLRP. This problem is then solved using a newly developed metaheuristic approach (a non-interactive multi-objective method), which attempts to obtain the full set of non-dominated solutions. Corresponding computational results are presented (using a set of benchmark instances from the single-objective CLRP literature), and a graphical example is analysed.

In Chapter 5 a different approach to the previously defined multi-objective CLRP is made. In this chapter, rather than attempting to obtain the full non-dominated set, only the non-dominated solutions considered to be interesting to DMs are attempted to be generated. This leads to interactive multi-objective methods, which are reviewed (focusing on open communication protocol), and a new proposal is made. The new proposal uses an open communication protocol to interact with DMs in order to obtain non-dominated solutions. The new interactive multi-objective method can also be used on other MOMIP problems. The method is applied to a specific multi-objective CLRP, where a step-by-step example is provided.

All the aforementioned approaches are most effective at improving decision support when integrated in a DST. The main goal of Chapter 6 is to present development methodologies and key success aspects of decision support systems. For the correct development of these information systems, several aspects have to be considered regarding: stakeholders involved, main development activities, development methodologies, and human computer interaction. These are studied in this chapter, where the development of a DST for solving LRPs is also analysed, with main phases of the adopted software development process being presented.

In Chapter 7, the DST developed for LRPs (following the main development phases as described in Chapter 6) is presented. Main functionalities are briefly described and the graphical user interface (namely data input and visualization features) is evaluated using usability testing. The tool is able to obtain results not only for LRPs but also for facility location and vehicle routing problems.

In the last chapter it is presented an overview of the main conclusions of this thesis, summarising the developed work and corresponding limitations. Future work and promising research directions are also identified.

Chapter 2

An Overview of Location and Routing

When addressing logistics systems several activities have to be taken into consideration. Some of these activities are handled on a daily basis while others are tackled on a medium or long term perspective. Although they refer to different time horizons, they may be linked, as the former corresponding decisions are typically biased by the latter (and vice versa).

In this chapter an introduction and an overview (mainly focused on mathematical approaches) are presented for some of the main decisions in logistics systems (namely facility location and vehicle routing), and a discussion concerning the need to tackle these problems in an integrated fashion is done. Afterwards, a taxonomy is proposed for the integrated approach (location-routing) (Lopes et al., 2008b).

2.1 Location and Routing Decisions in Logistics Systems

A logistics system is composed of a set of facilities and final users, where products are to be distributed using transportation services (Ghiani et al., 2004). The term facility (or depot) is usually used in its broadest sense, as it can refer to factories, schools, warehouses, distribution centres, hospitals, retail outlets, post offices, dump sites, to name but a few. The final users can be consumers, communities, companies, or even points of demand and/or supply and are often named clients. Transportation services, via routes or paths, regard moving materials between facilities (and final users) using vehicles. Routes are vehicle tours passing through more than one client or depot, before returning to the departure point. Paths include not only routes in which vehicles do not return to the departure point but also direct links between origin and destination points.

The role of logistics systems models is to support the decision maker (DM) in choosing the system which will provide the best combination of cost and service among the possible alternative configurations. Min and Eom (1994) and Goetschalckx et al. (2002) advocate that these models can play an important role in identifying and evaluating alternative courses of logistics actions. This is due to their ability to structure complex managerial goals, constraints, and variables with enormous accuracy. However, their usage could be jeopardized if they do not successfully link interrelated logistics activities and establish an integrated view of the system (Min and Eom, 1994). Therefore, correctly modelling logistics systems can prove to be a daunting task. The high degree of complexity of these systems, where a large number of components exist, usually with complex interrelationships, suggests a systems (namely, systems engineering) approach as the best way to develop logistics models (House and Karrenbauer, 1982).

Within systems engineering the use of several tools and methods to better comprehend and manage complexity is encouraged (Blanchard, 2008). One of the main methodologies, which has often been employed in the development of logistics models, is optimization (Ghiani et al., 2004). From this field of applied mathematics, the focus of this thesis will be combinatorial optimization where, generally, (mixed) integer programming models are used to formally describe problems. However, in combinatorial optimization, problems generally belong to the NP class and, as such, even the simplest of logistics systems models cannot be solved as a whole in reasonable time. This has led to the separate study of the several components (or activities) of logistics systems (Daganzo, 2005). Still, in order to provide a model that can portray reality to some extent, a correct partition is required.

Several categorizations for the range of activities (and corresponding decisions) within a logistics system have been proposed (Riopel et al., 2005). The most common categorization groups activities into three hierarchical levels, depending on the scope, time horizon, and frequency (Perl and Sirisoponsilp, 1988): strategic, tactical, and operational.

Strategic decisions are generally related to major and more expensive logistics aspects which are to be considered over relatively long periods of time, and as such, are hardly reversible and have long-lasting effects. Tactical decisions are typically related to moderate investments, to be made on shorter time frames (annual, semi-annual, or seasonal time horizon), and can usually be reverted at a moderate cost. Finally, operational decisions are generally made on a daily basis or in real-time and are characterized by low investments, which can be reversed without incurring significant costs.

Examples of some logistics decisions (facility location, transportation, and inventory) categorized by the three aforementioned hierarchical levels can be seen in Table 2.1.

Table 2.1 Classification of facility location, transportation, and inventory decisions into three hierarchical levels (Perl and Sirisoponsilp, 1988).

Logistics decisions	Strategic	Tactical	Operational
Facility location	Number and location of facilities	Material handling equipment	
	Assignment of facilities to supply sources		
	Allocation of demand to facilities		
Transportation	Mode	Carrier	Assignment of crew and loads to vehicles
	Type of carriage	Shipment size	Routing/scheduling
Inventory	Total systems inventory	Size of inventories at various locations	Control discipline at various locations
	Location of inventories	Levels of safety stock at various locations	

The different decision levels of logistics activities have led to a somewhat natural separation of the problems, as practitioners try to adapt models to specific logistics decision environments and time frames. However, as mentioned earlier, this may not be the most correct approach as it lacks the interrelation between activities.

Albeit all logistics activities are of the utmost importance, the objective of this work is to focus on facility location and vehicle routing, two components of logistics systems that tend to be closely linked.

It can easily be concluded that they refer to two different kind of decisions (see Table 2.1): strategic (location) and operational (routing). For this reason, combined with the inherit complexity of the two problems, they have often been tackled separately, which led to several distinct models in the literature. The correct planning of these activities can result in a significant improvement being currently a critical success factor for many organizations and, in order to achieve this, an integrated view of these problems is required.

In the following subsections an overview of several existing models for these two components of logistics systems will be made, to finally address the integrated approach, named location-routing.

2.1.1 Facility Location

When choosing potential facility locations, DMs have in mind several aspects. Among these we have physical and economic constraints, but also motivations regarding future company needs, service levels, environmental aspects, company policies, and so forth (Tompkins et al., 2010). Nevertheless, when such a problem is addressed, often there is a preliminary study, in which the DM has narrowed the choices to a subset of potential locations compliant with the system's specification.

Based on the subset of candidate sites it is necessary to determine which is the most suitable regarding a given objective. This objective can be, for example, cost minimization, coverage maximization, obnoxious effect minimization, optimization of some equity measure, a combination of the above, or even, it can be desirable to simultaneously consider several different opposite objectives (Eiselt and Laporte, 1995).

Facility location problems can thus be defined as intended to determine the optimal location of a fixed or variable number of facilities, with respect to some economical or social measure, while guaranteeing a predetermined service level (Albareda-Sambola, 2003).

Currently there are several surveys regarding facility location problems (Daskin, 1995; Drezner and Hamacher, 2002; ReVelle and Eiselt, 2005; Melo et al., 2009). Based on the previously assumed DM's position (where there is only a subset of potential facility locations) this work will focus mainly on discrete facility location problems as opposed to continuous and network facility location problems where, respectively, the facility can be located anywhere on a d-dimensional space $\mathbb{R}^d$, $d \in \mathbb{N}$ (usually $d = 2$, and is named planar location), or the feasible region is a network (note that discrete problems can be considered as a particular case of network problems).

Much of the literature on facility location modelling has been directed at formulating new models and modifications to existing ones, rather than directing at specific applications (Current et al., 2002). This has led to the definition of a set of basic problems from which adaptations can be later made to tackle specific scenarios. As follows, some of these problems will be presented, where the underlying network is given, as well as the location of the demand points to be serviced by the facilities.

Set Covering Problem

In the set covering problem, first present by Toregas et al. (1971), the objective is to locate the minimum number of facilities required to cover all of the demand nodes. The concept of "cover" relates to a predefined maximum distance to the facility, to which, if the client is within, is considered covered (fully satisfied). Moreover, a client can only be satisfied or not satisfied, meaning that being closer than the maximum distance does not improve satisfaction.

Maximal Covering Location Problem

The maximal covering location problem (Church and ReVelle, 1974) differs from the set covering problem in having an upper limit on the number of facilities to install. Hence, the objective is to locate a predetermined number of facilities (p), in order to maximize the covered demand. Unlike the set covering problem, there may be unmet demand.

***p*-Centre Problem**

The previous location problems assume that the covering distance is a fixed and predetermined standard, however it is often not the case. In the p-centre problem (Hakimi, 1964), it is required to minimize the maximum distance of all demand nodes to its closest facility constrained by a given number p of facilities. In this problem, where equity is sought, there is no longer a predefined standard maximal covering distance.

***p*-Median Problem**

In the p-median problem (Hakimi, 1964) it is intended to find the location of p facilities that minimizes the sum of demand-weighted distances between each client and the facility to which is assigned. Instead of the p-centre equity objective, here the goal is efficiency.

Fixed Charge Location Problem

In this problem, formulated by Balinski (1965), the objective is to minimize the fixed facility location costs and the total travel costs required for demand to be serviced. Unlike the previous problems, however, there is a limit associated with the capacity of each facility, which may have different installation costs, and there is no *a priori* limit on the number of facilities to open.

Maxisum Location Problem

Previous problems assume that it is advantageous to locate facilities as close as possible to demand points. However, when dealing with undesirable facilities (e.g. power plants, prisons and landfills for hazardous wastes) at least one of the objectives involves locating facilities as far as possible from demand nodes. The maxisum location problem (Church and Garfinkel, 1978) handles the location of p facilities such that the total demand-weighted distance between demand nodes and assigned facilities is maximized (as opposed to the p-median minimization objective).

2.1.2 Vehicle Routing

A large part of many logistics systems involves the management of a fleet of vehicles used to service facilities, retailers and/or clients. In order to control the costs of operating a fleet and meet the required level of service, it is necessary to continuously make decisions on how much to load on each vehicle and (from) where to send it (Bramel and Simchi-Levi, 1997; Rushton et al., 2006). As previously stated, these are operational activities that often require daily analysis leading to being one of the most studied in combinatorial optimization.

From these problems it can be identified scenarios in which vehicles can supply more than one client (also named “less than a truck-load”). Furthermore, one can distinguish between problems with one vehicle, serving the totality of clients, and problems in which capacity constraints force the use of several vehicles. The former can be tackled using a travelling salesman problem (TSP), where the objective is to find the shortest possible route that visits each client exactly once (Applegate et al., 2006); while the latter fall under the general class of vehicle routing problem (VRP), which will be addressed hereafter.

The VRP is one of the most challenging combinatorial optimization task, falling into the category of NP-hard problems (Garey and Johnson, 1979), and can be defined on a graph with a vertex serving as a depot and travel costs or times associated with each arc. The objective consists in designing the optimal set of routes, for a fleet of vehicles, in order to service a given set of clients (located on the remaining vertices). The VRP has led to a variety of problems as well as closely related ones (comprehensive reviews and some applications can be found in Crainic and Laporte, 1998, and Toth and Vigo, 2002). Some of these problems will be addressed as follows.

Capacitated Vehicle Routing Problem

The capacitated VRP (CVRP), introduced by Dantzig and Ramser (1959), is a VRP in which a fleet of vehicles, of uniform capacity, must service a known set of clients demand for a single product from a common depot, at minimum transit cost. What basically differs the CVRP from the VRP is the additional constraint that every vehicle must have uniform capacity of a single product. The objective is thus to minimize the number of vehicles and the sum of travel time.

Multi-Depot Vehicle Routing Problem

It is common for a company to have several depots from which it can service its clients. If the clients are heavily clustered around the depots the problem can be reduced to a set of independent VRPs. When it is not the case it should be assumed as a multi-depot VRP, where it is required to assign clients to depots. In each depot, a fleet of vehicles is based which departures from the depot, services the assigned clients, and returns to the same depot. A formulation for this problem can be found in Perl and Daskin (1985).

Vehicle Routing Problem with Time Windows

Often, the distribution of products is associated with an interval of time wherein the client has to be serviced. These problems are named VRPs with time windows (Cordeau et al., 2002) and differ from the VRP in having a time window associated with each client (at the depot the interval is called the scheduling horizon). Moreover, the objective adds, to the usual vehicle fleet and travel distance minimization of the VRP, the minimization of the waiting time needed to service all clients in their required hours.

Vehicle Routing Problem with Pickup and Delivery

This problem is a VRP in which it is contemplated the possibility of clients returning some products (Parragh et al., 2008b). Therefore, it has to be taken into account the products the clients want to return to the delivery vehicle, so they can fit into it. This constraint can lead to a poorer utilization of the vehicles capacity, increased travel distances or the need for more vehicles, further increasing the problem difficulty.

Vehicle Routing Problem with Backhauls

Similarly to the previous problem, in the VRP with backhauls (Parragh et al., 2008a), clients can demand or return some products. The main difference is, in this problem, all deliveries must be completed before any pickups can be made. This arises from the fact that often vehicles are rear-loaded and rearrangement of the loads during the route is not deemed economical or feasible.

Arc Routing Problems

These problems can be seen as closely related to the VRP (Dror, 2000). Likewise, they can be defined on a graph with a differentiated vertex (the depot), and costs associated with the arcs. Unlike the VRP however, the clients, rather than being on the vertices, are along the arcs, requiring them to be visited (all or a subset of the arcs) at least once. Most of the variants of the VRP are applicable to arc routing problems, being the most known the capacitated arc routing problem (CARP) the CVRP arc routing counterpart.

2.1.3 Location-Routing

The facility location problems addressed earlier consider either the service to be performed in the facility site (requiring clients to travel there) or a dedicated trip required from the depot to each client. VRPs, although allowing to service several clients with a single trip, are already biased by the location of the depot(s).

Most of the existing mathematical models focus on these two components of the logistics system individually, however, location and distribution are two components that tend to be closely linked. In many real-world situations, it is necessary to install one or more facilities (or services) and, simultaneously, establish the distribution between facilities and customers. The traditional approach of locating first and then designing the distribution routes has been, in these cases, gradually replaced by an integrated approach of the location and routing (Balakrishnan et al., 1987). Salhi and Rand (1989) consider that these components are strongly linked and should not be optimized separately.

The integrated approach has been named location-routing problem (LRP) (Laporte, 1988) and aims to model and solve facility location problems, while creating the distribution routes. LRPs typically combine three different decisions: the number, size and, location of facilities; the allocation of the demand points to the facilities; and the design of the vehicle routes emanating from the facilities.

Location-routing encompasses a set of problems within location theory. Balakrishnan et al. (1987) observe that LRPs are strategic decisions concerning facility location; Nagy and Salhi (2007) state that these problems aim to solve facility location problems simultaneously solving a VRP. Overall, these can be seen as strategic problems that intend to determine the location of facilities taking into account distribution aspects.

The distribution aspect of LRPs can be generalized in order to include all types of vehicle distribution considerations, be it by using either routes or paths. Other approaches can also be used in the routing aspect of LRPs, such as route length estimation, in which the actual routing is substituted by a cost (or a surrogate like length) estimation (Laporte and Dejax, 1989; Chien, 1993; Nagy and Salhi, 1996b; Bruns et al., 2000; Shen and Qi, 2007).

Moreover, the only distribution considered in LRPs is “offline” routing (knowledge about the environment in which routing takes place is available beforehand). For “online” (or real-time) routing it becomes a queueing location problem (reviewed by Berman and Krass, 2002).

The LRP can also be identified by looking at the desired physical structure (Laporte, 1988). Figure 2.1 exemplifies some possible situations (which will be addressed later) where both location and distribution are addressed at several levels. From the presented problems (Figure 2.1) only the location-allocation problem is not considered to be an LRP, due to having no routes/paths between supply (depots) and demand points (clients), meaning no true distribution decisions are considered.

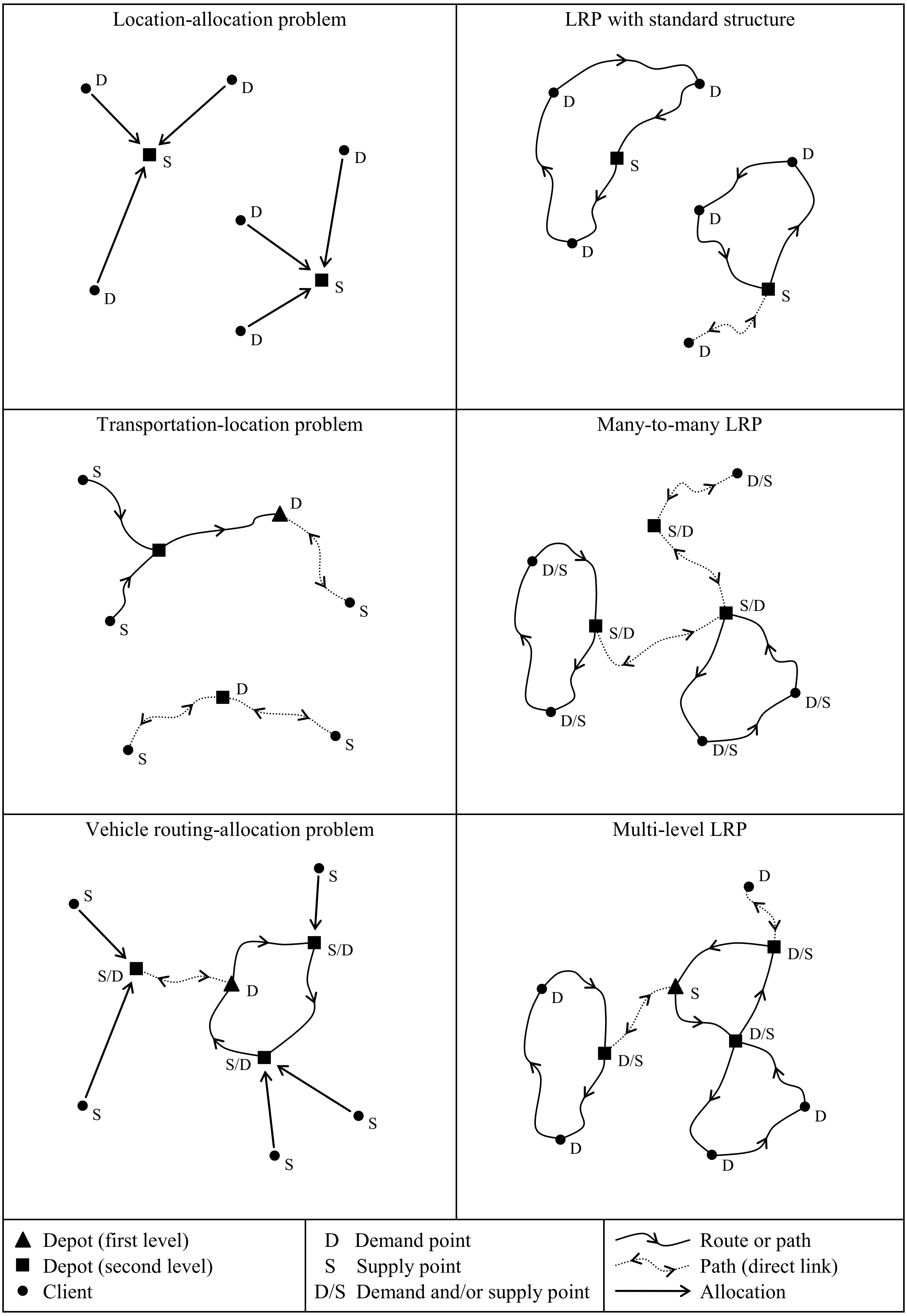


Figure 2.1 Graphical representation of some problems regarding location and distribution.

Other very similar types of problems usually not considered as LRPs are the generalized network design problems – like the generalized TSP (Laporte and Nobert, 1983) or the generalized VRP (Ghiani and Improta, 2000). These can be defined as follows: given a depot and several sets of nodes, it is intended to determine the Hamiltonian circuit through the depot and at least one node from every set. Although dealing with node selection and sequencing on a tree, they are mainly directed at problems with null (or next to null) depot installation costs (node selection). Having no fixed cost, their precise locations on the various sets of nodes are immaterial, therefore, not dealing with true location issues but rather distribution decisions (e.g. transhipment). Nagy and Salhi (2007) briefly discuss other related problems.

2.2 Taxonomy of Location-Routing Problems

By now, it is common knowledge that location and routing are interrelated (Salhi and Rand, 1989). However, both practitioners and academics often ignore this when approaching integrated logistics problems on locating facilities. Many practitioners albeit being aware of this, still tend to search for a separate answer (Rand, 1976). According to Nagy and Salhi (2007), there are some possible explanations for this behaviour: the practical situation does not require routing aspects; the researchers share a view in which these are two different levels of decision (location being a strategic decision with a long-term planning horizon, while the routing component is mainly operational with short-term planning objectives, hence more inclined to be recalculated and redraw on a more regular basis); and the LRP is conceptually more difficult than the location problem, making the latter easier to manage.

Although some of these arguments have been refuted in the literature (Salhi and Nagy, 1999), it is possible, nevertheless, to identify situations where it may be even more important to find solutions using this type of problems: when routing has a significant impact, be it cost wise (in many product distribution application) or due to the nature of the transported products (e.g. dealing with the transportation of hazardous material – HAZMAT).

There has been a significant thread of works regarding the LRP and it may be important to segment them in order to find gaps and foster new research. In this section a survey will be made with this purpose: providing a new proposal for a taxonomy of LRPs; updating and adding other less known (and somewhat neglected) works; reviewing the approaches adopted in the literature; and addressing a common issue in the day-to-day life of many of these decisions which is the multi-objective problem.

Firstly, it will be discussed the methodology adopted for the taxonomical classification. Then, the taxonomy primarily based on the problems intrinsic characteristics will be addressed. Finally, it will be presented an overview of the current algorithmic approaches and main objectives. A brief analysis of the classified papers can be found in Appendix A.

2.2.1 Methodology

In the development of the presented taxonomy, a review of several surveys and taxonomies addressing the LRP was made (Madsen, 1981; Laporte, 1988, 1989; Min et al., 1998; Nagy and Salhi, 2007). Some of the issues raised in these papers are also addressed in this work. It is intended to propose a taxonomy that can be widely accepted and can cope with all current LRP works as well as flexible enough to easily adjust to future research.

The focus will be on journal articles published (or available online) in major English language publications as this is the current *lingua franca* of the academic world. Consequently, working papers, conference proceedings, book chapters, master and doctoral theses, and non-English articles regarding this issue are not presented here. This exclusion derives from the specific nature of these works which are not generally disseminated. Decisions regarding inclusion of articles in any literature review are somewhat arbitrary and reflect biases and special interest of the reviewer. This taxonomy tries to be as unbiased, exhaustive, and extensive as possible.

Due to the diversity of characteristics of location-routing models, there are several approaches for this type of problem, generally corresponding to real-world needs and according to the addressed scenario. Table 2.2 shows a set of (default) characteristics that can be considered when tackling an LRP.

Table 2.2 Default and possible characteristics of an LRP.

		Default	Other possible characteristics	
Depots	Number	Maximum allowed	Fixed	
	Capacity	Capacitated	Uncapacitated	
	Type	Homogeneous	Heterogeneous	
	Cost	Fixed	Variable	
	Service provided	Desirable	Semi-obnoxious	Obnoxious
	Candidate sites	Finite set (discrete)	Infinite set (continuous)	
Clients	Operation	All deliveries or all collections	Mixed deliveries and collections (only one)	Both deliveries and collections (per client)
	Nature of demand	Deterministic	Stochastic	
	Location	Nodes	Edges	Mixed
	Serviced	Once	Several trips	
	Service schedule	No constraints	Fixed time	Time windows
Vehicles	Number per depot	Several	One	
	Type	Homogeneous	Heterogeneous	
	Capacity	Capacitated	Uncapacitated	
	Cost	Fixed	Variable	
	Assigned routes	One	Several	
	Covered distance	No constraints	Maximum allowed	Equity
	Route type	Route (ends at the departure point)	Path (does not end at the starting point)	Path (direct link)
	Route time	No constraints	Maximum duration	Time windows
Products	Number	One	Several	
	Characteristics	None addressed	Hazardous	Volume

Based on the diversity of these characteristics the task of addressing a complete characterization of the models may represent a challenging and difficult task. In order to surpass this, the articles were grouped according to the major concerns addressed, always taking into consideration that there may exist other specific issues that differ among works inside the main categories. This somewhat simplified taxonomy focuses primarily in the models topology and is composed of two levels (presented in Figures 2.2-2.16).

In the first level of the proposed taxonomy, the algorithmic approach is not taken into consideration, but rather the physical aspects of the models. The decision to initially categorize the articles this way was based upon the desire to create a general view of the LRP, to foster future research on the several proposed segmented areas. Classification decisions, at this level, were based upon the underlying purpose of the formulation, that is, the physical characteristics of the studied problems.

Later, a second level of the taxonomy, encompassing the algorithmic approach and the objective focus is provided. This final level is composed of two tiers. The first is based upon the solution technique which separates into two broad categories: exact solution techniques and heuristic techniques. The second tier separates the articles according to the number of objective functions (single-objective or multi-objective). Although further classification may be possible, it would eventually add unnecessary complexity to the classification.

2.2.2 Taxonomy and Classification

The main focus of this taxonomy is to group the papers in the literature according to its topology, trying to encompass all the works considered as LRPs. Further works may afterwards be incorporated in this classification, possibly even extending it horizontally.

The admittedly arbitrary decision to create this segmentation derives from the need to create smaller topological groups and is in line with the recent review of Nagy and Salhi (2007).

As previously stated, this taxonomy is mainly oriented to the physical characteristics of the problem. In order to segment the different models, a first separation according to the main features of common LRPs was considered (Figure 2.2). Thus, in a first layer, the hierarchical structure of most LRPs, which consists of clients and depots (with possible several depot levels), in which depots service a number of clients by means of vehicle routes, was addressed. The mainstream research focuses on this topology, however, some stray from this characterization, be it by considering routing at several levels (e.g. routing in-between depots), or even by using paths instead of routes. The former are categorized as [1] while the latter are labelled [2] (see Figure 2.2).

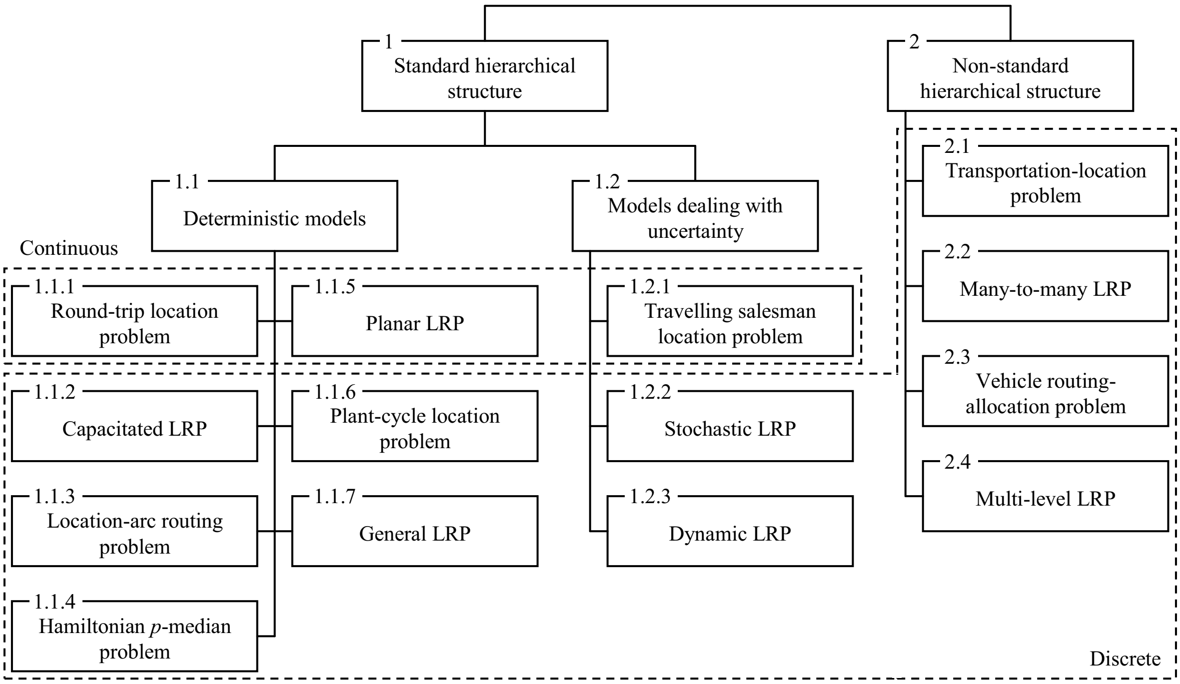


Figure 2.2 First level of the proposed LRP taxonomy.

Going further into the differences between LRP approaches, a new separation of the problems considered to sustain a standard hierarchical structure [1] can be made. In this second layer it is possible to split the problems according to the perceived certainty. On one hand, studies where there is a significant knowledge of the situation at hand (a predictable or static environment exists), in which case the problem is approached using deterministic data and assuming a single period of time (generally named static problems) [1.1]. On the other hand, group [1.2] copes with problems where there is a degree of uncertainty being reflected into the model by either performing several analyses over time (dynamic problems) or by incorporating stochasticity into the model (typically, clients demand).

The final layer of this first classification groups the problems into somewhat defined distinct models in the literature (e.g. round-trip location problem, travelling salesman location problem, and transportation-location problem) as seen in Figure 2.2.

Moreover, a typically important characteristic of location problems is displayed: continuous or discrete location. Some of these problems work with location (anywhere) on a plane, with an infinite set of possible depot location (continuous), in opposition to the usual finite set (discrete) approach, where a set of previously selected candidate sites for depots exist.

The second level of the taxonomy (further addressed in Section 2.2.3) intends to classify the papers according to the adopted approach. Firstly, the classification is based on the solution method dividing into exact and heuristic techniques. Secondly, the objective function(s) is(are) tackled by presenting the problems classified according to single-objective (when a single objective is considered) or multi-objective (where several opposite objectives are simultaneously addressed).

However, some papers address several issues or even algorithms, and were categorized according to what can be considered as the most defining aspect of the work.

As follows, each different type of problem will be briefly described and some examples of possible real-life applications are presented.

Round-Trip Location Problem [1.1.1]

Vehicles start out from a depot, pick up a cargo from a client and deliver it to another client, then returning to the depot. A common application is the determination of the location of a courier service. Some of these problems work with continuous location.

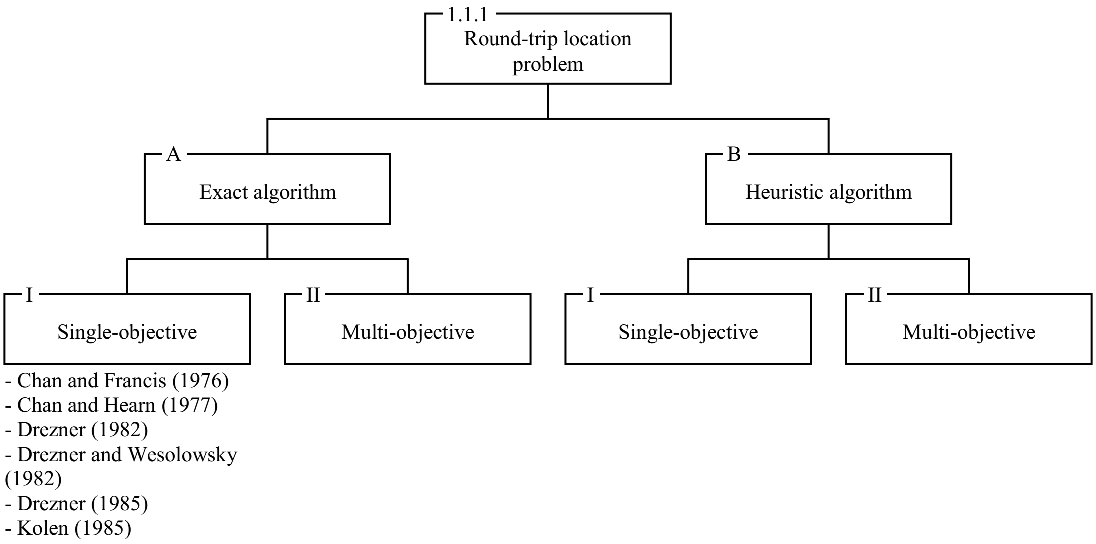


Figure 2.3 Round-trip location problem branch of the proposed taxonomy.

Capacitated Location-Routing Problem [1.1.2]

From the mainstream of LRP studies a problem with specific characteristics has emerged. This is the capacitated LRP (CLRP) where only two levels are considered (clients and depots) and the only route constraints are regarding the vehicle capacity (a fleet of identical vehicles with homogeneous capacity of a single product is assumed). Moreover, a capacity may be assigned to each depot. This can be seen as an extension to the CVRP and several real-world scenarios may fit this definition (e.g. determining the location of a production facility which services a number of clients).

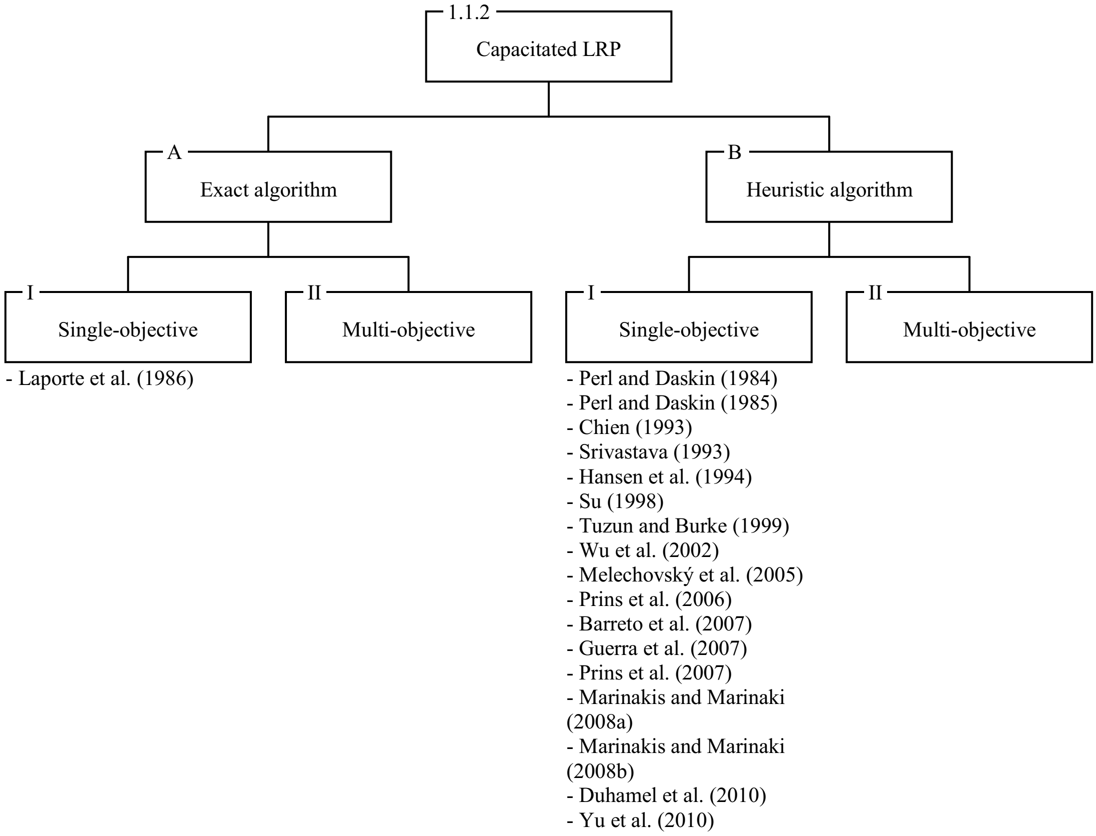


Figure 2.4 CLRP branch of the proposed taxonomy.

Location-Arc Routing Problem [1.1.3]

The location-arc routing problem (LARP) encompasses problems in which the clients instead of being on the nodes of the networks are on the arcs (i.e. Euler cycles are assumed). Possible real-world applications would be situations where it is intended to determine location when the demand is throughout the arc. Examples of such scenarios include locating facilities for postal delivery, garbage collection, road maintenance, winter gritting, and street sweeping.

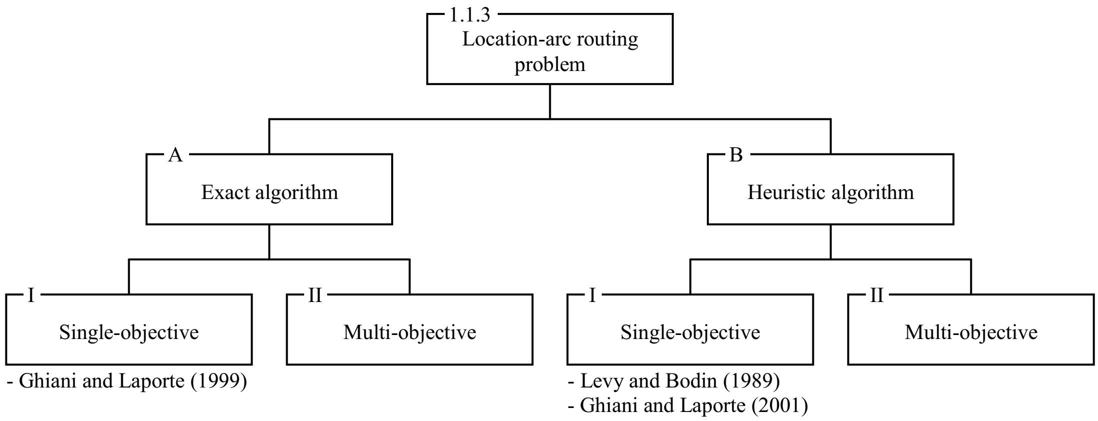


Figure 2.5 LARP branch of the proposed taxonomy.

Hamiltonian *p*-Median Problem [1.1.4]

It is intended to locate exactly *p* depots (in which the number of depot locations can correspond to the number of clients, being each client a potential depot site) and each depot has exactly one vehicle. The problem embeds the *p*-median problem (by finding the least cost partition for *p* depots) and the TSP.

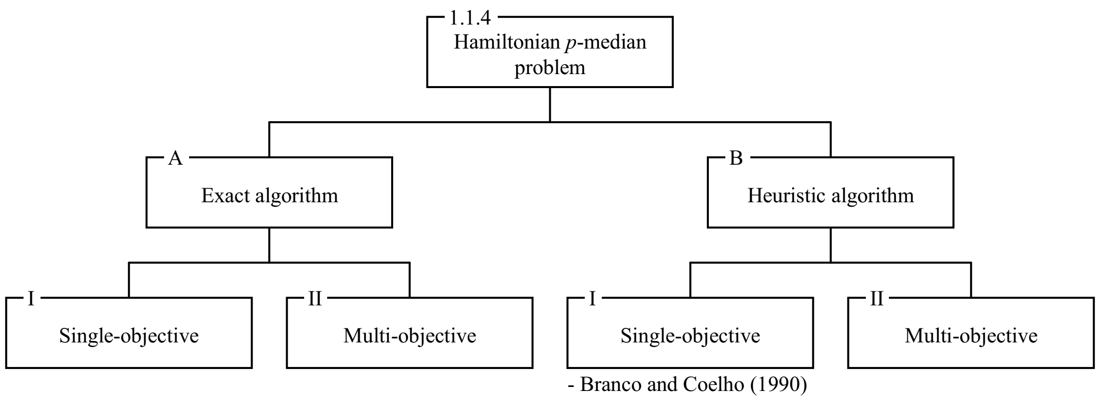


Figure 2.6 Hamiltonian *p*-median problem branch of the proposed taxonomy.

Planar Location-Routing Problem [1.1.5]

In these problems rather than discrete location, continuous location is considered. Although the road network is generally assumed as discrete there may exist situations where the location of a facility may not necessarily be on a road (but on the plane instead). As some works on the round-trip location problem [1.1.1], these problems deal with an infinite set of possible locations (depots can be established in a continuous space, usually on the plane).

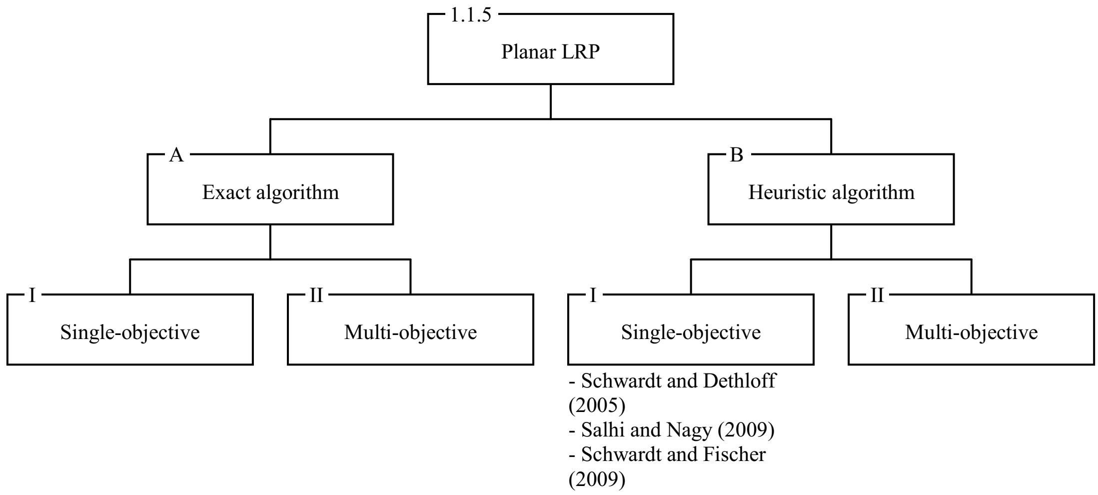


Figure 2.7 Planar LRP branch of the proposed taxonomy.

Plant-Cycle Location Problem [1.1.6]

It is simultaneously considered the location of stations and the design of (optical fibre) rings connecting radio antennae to the stations. In this problem instead of vehicle routes it is addressed communication rings (making the routing decision less at the operational level and more at a strategic level due to unlike vehicle routes, where often a road network with alternative routes is available, once the communication rings are established between the radio station, significant costs may be incurred to alter the established ring). Telecommunications antennae determination is an obvious application for these problems.

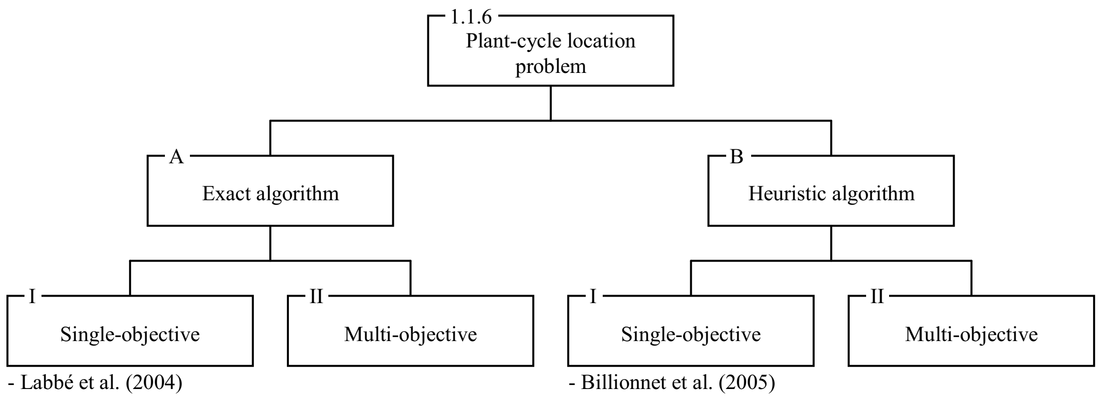


Figure 2.8 Plant-cycle location problem branch of the proposed taxonomy.

General Location-Routing Problem [1.1.7]

This category encompasses general deterministic problems that are not incorporated in the previously mentioned classes, including some works where the standard LRP is modified to model specific scenarios. These represent several usual applications, for instance, determining the location of an intermediary facility and necessary vehicle routing of a three level (two depots and a client levels) logistics system.

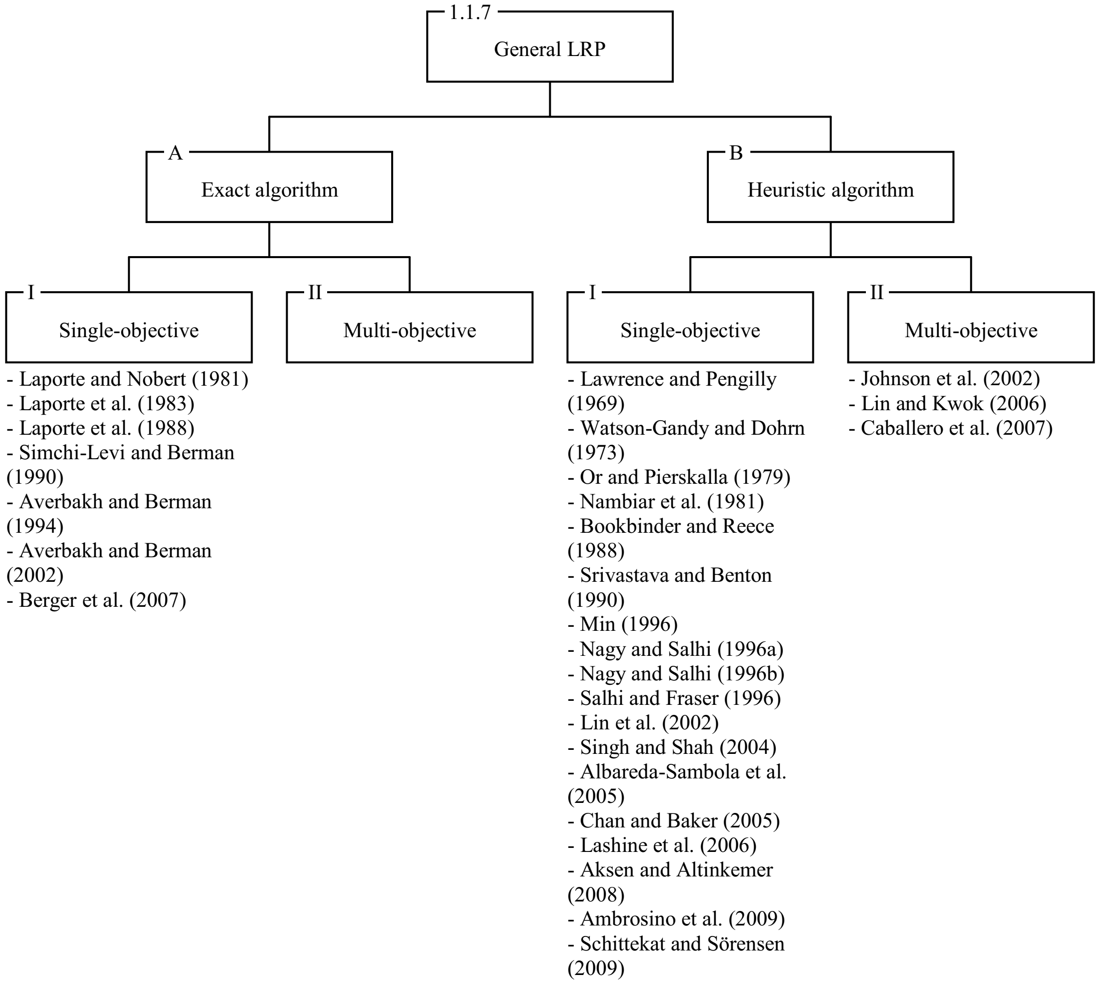


Figure 2.9 General LRP branch of the proposed taxonomy.

Travelling Salesman Location Problem [1.2.1]

This is one of the most addressed problems in the LRP literature where it is intended to determine the home base of a travelling salesman. It deals with a single depot and vehicle while the stochastic variation is concerning the customer demand (only a subset of the clients is randomly selected to be serviced). The only exception to this is the work by Berman and Simchi-Levi (1989) where the lengths of the links in the network are stochastic.

Some of these problems deal with location on the plane (continuous location or infinite set approach). In this LRP variant it can also be included the probabilistic TSP (using *a priori* tours: firstly a tour is constructed for all the clients; secondly each client that, for a given route, doesn't require service is skipped). Possible applications may consist in determining the location of a district sales office, or a tourist choice of the hotel to check-in in order to visit several places of interest.

Other extensions to this problem have also appeared in the literature: the delivery man location problem and the sales-delivery man location problem (in which the main difference lies in the sought objective); and the location of several travelling salesman. These variants however, have only considered the deterministic approach (the only exception is the work by Averbakh and Berman, 1995, for the probabilistic sales-delivery man location problem), and as such they can be found in [1.1.7.A.I] (Simchi-Levi and Berman, 1990; Averbakh and Berman, 1994, 2002).

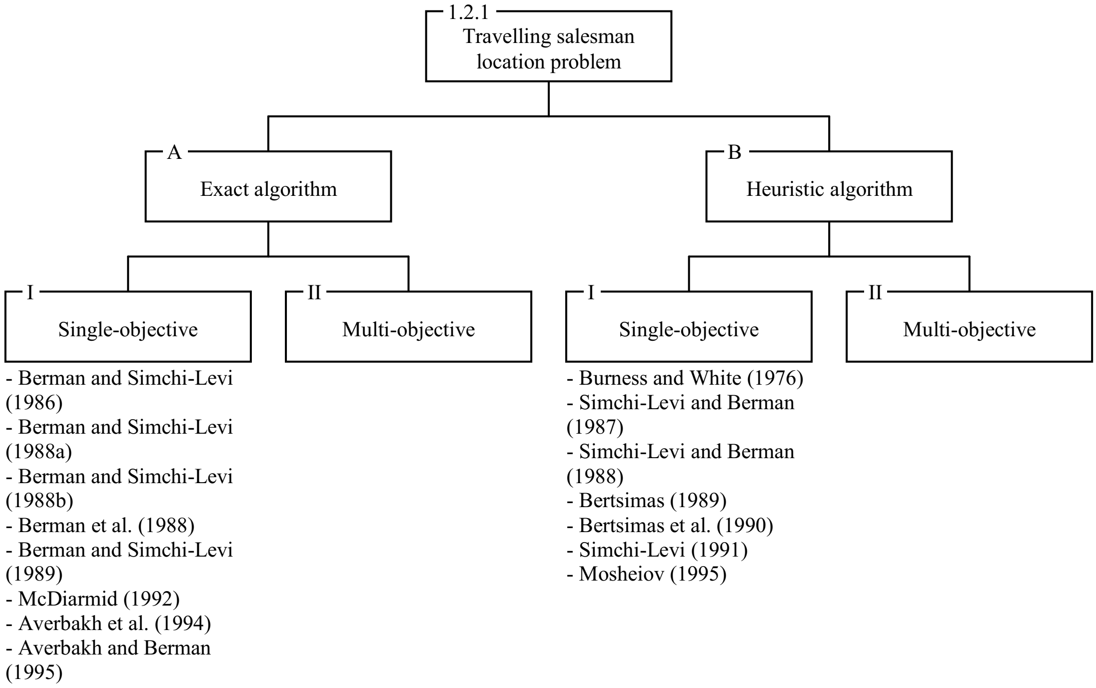


Figure 2.10 Travelling salesman location problem branch of the proposed taxonomy.

Stochastic Location-Routing Problem [1.2.2]

Addresses general stochastic problems not included in the previous category. These represent general LRP where stochastic data are considered (usually the clients demand). Practical situations may be depot installations where there is a degree of uncertainty regarding future demand or even when distributing goods with seasonal demand.

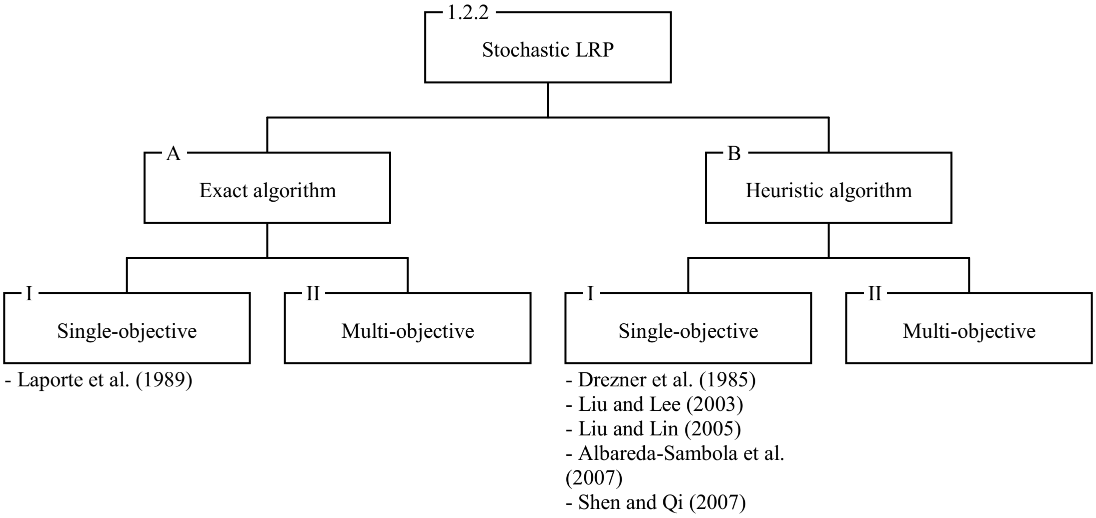


Figure 2.11 Stochastic LRP branch of the proposed taxonomy.

Dynamic Location-Routing Problem [1.2.3]

In these problems the location of depots over a planning horizon of several periods of time is studied, instead of the usual static approach. Similarly to the stochastic problems it is intended to insert into the model some degree of variation in order to provide more flexible solutions and to cope with uncertainty.

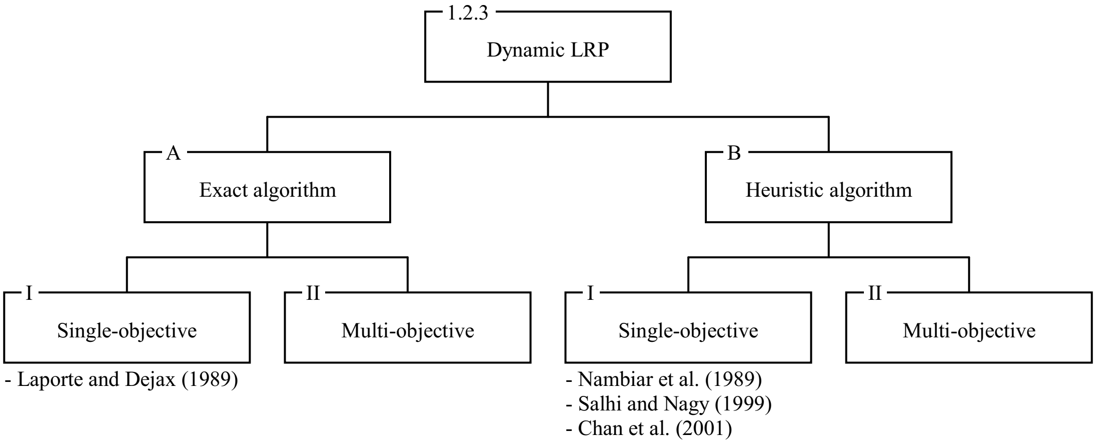


Figure 2.12 Dynamic LRP branch of the proposed taxonomy.

Transportation-Location Problem [2.1]

It deals with locating depots and finding paths instead of routes (direct links are usually considered). The path is, in most cases, between supply and demand points. This is frequent in the transportation of hazardous/nuisance materials, in which falls upon most papers in this category. Another example is the location of hospitals where it is crucial not to have stops for each ambulance run and transportation to adjacent communities must be considered. A special case of this problem, referred in the literature as the transhipment location problem, is when the facility to locate is between the origin-to-destination paths, typically involving cargo transhipment.

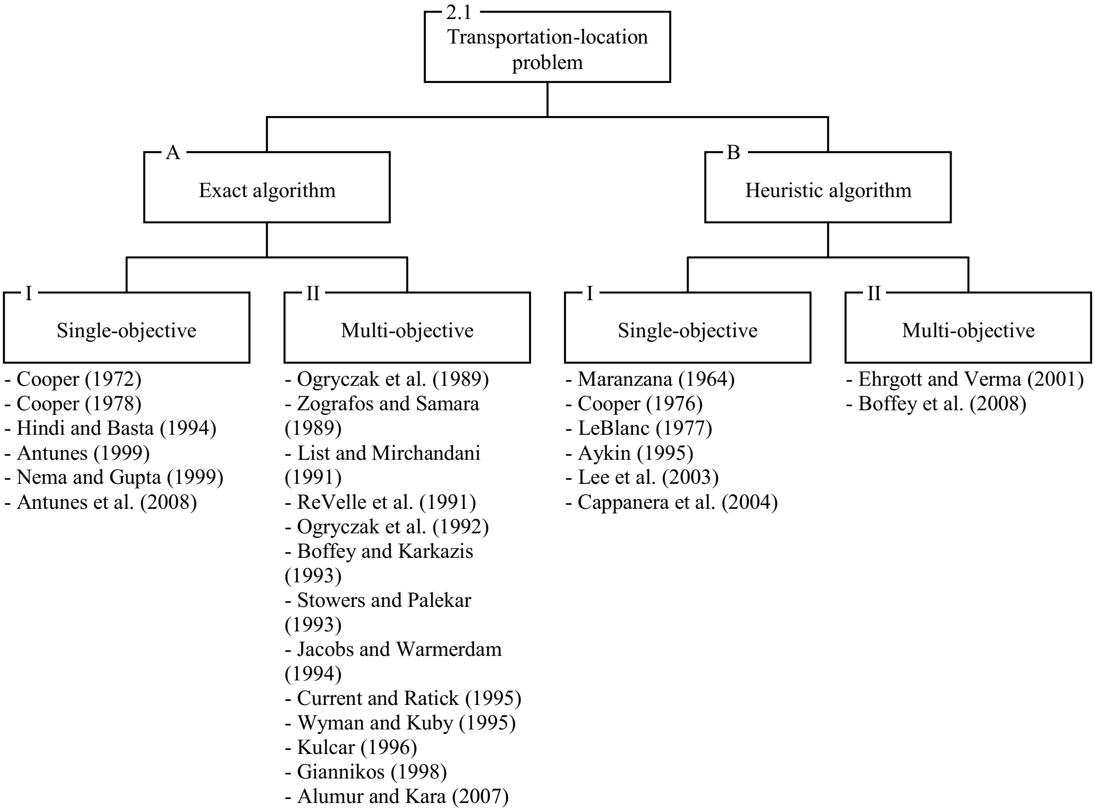


Figure 2.13 Transportation-location problem branch of the proposed taxonomy.

Many-to-Many Location-Routing Problem [2.2]

In this problem, introduced by Nagy and Salhi (1998), it is intended to locate depots where several clients wish to send products to each other (meaning all clients potentially have pickup and delivery). A real-world scenario would be the postal flow between communities or even freightliner terminals for road transportation. Moreover, in-between depots it is only assumed to exist direct links and different routing costs may exist at both distribution levels (inter-depot and depot-to-client).

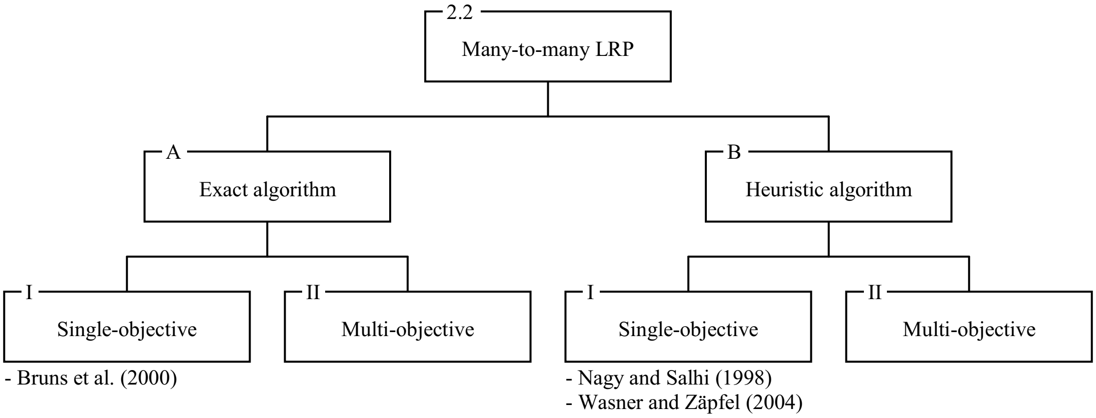


Figure 2.14 Many-to-many LRP branch of the proposed taxonomy.

Vehicle Routing-Allocation Problem [2.3]

This category addresses problems where routing is inter-depot (at the depot level) instead of the most common depot-to-client (client level). This is the case when clients have do make a (typically small) trip to the depot, like in the determination of the location of post boxes. This problem was firstly presented and formulated by Beasley and Nascimento (1996)[1].

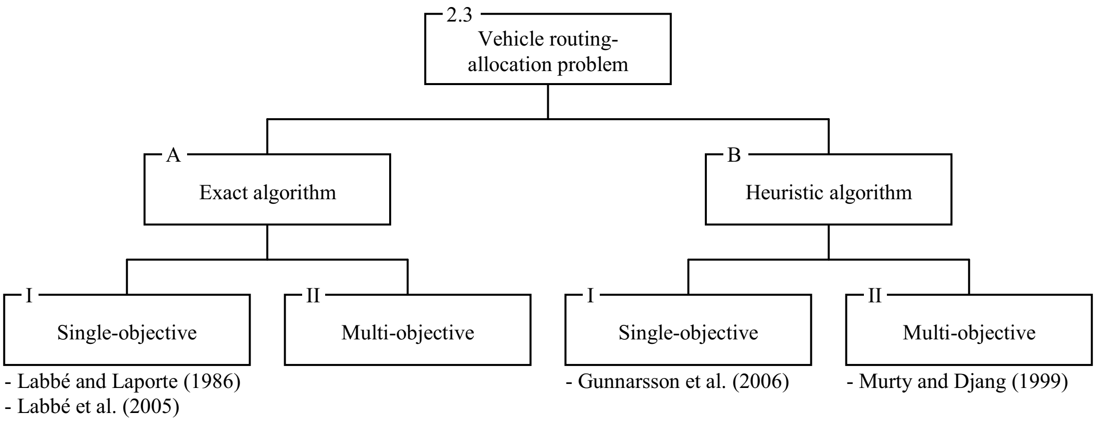


Figure 2.15 Vehicle routing-allocation problem branch of the taxonomy.

[1] The authors did not present an algorithm for solving the problem, reason why the work is not included in the proposed taxonomy.

Multi-Level Location-Routing Problem [2.4]

This problem tackles routing at both the depot and client levels. A practical example may be the distribution of newspapers. Firstly the distribution is made between the factory and the transfer points and finally from these to the clients. This category also includes the road-train routing problem which contains similar characteristics.

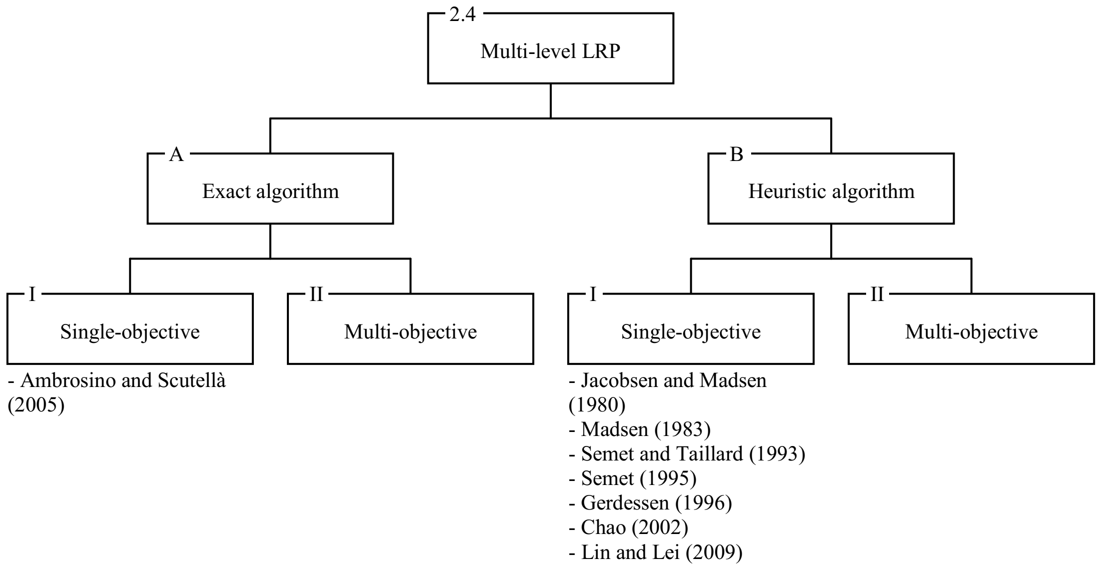


Figure 2.16 Multi-level LRP branch of the proposed taxonomy.

2.2.3 Algorithmic Approaches and Objectives

In this section, a listing of the different algorithmic approaches and used objectives found in the proposed taxonomy is presented.

Algorithmic Approaches

The second level of the taxonomy tends to classify the problems according to the adopted approach. The first rank in this classification is based on the solution method and is divided into exact and heuristic techniques. In some articles, however, both techniques are considered. When that is the case, the article is considered only once, inside the exact category (e.g. Cooper, 1978, Berman and Simchi-Levi, 1989, and Laporte and Dejax, 1989).

Exact algorithms in the LRP literature can be categorized as follows (in squared brackets is the problem classification and the biggest solved instances – [problem: clients x potential depots]):

- branch-and-bound [1.1.7: 80x3] (Laporte et al., 1988) [1.2.2: 30x3] (Laporte et al., 1989)

- branch-and-cut [1.1.3: 200x50] (Ghiani and Laporte, 1999) [1.1.6: 120x30; 100x100] (Labbé et al., 2004)
- branch-and-price [1.1.7: 100x10] (Berger et al., 2007)
- cutting-plane method [1.1.7: 50x15; 40x40] (Laporte et al., 1983)
- dynamic programming [1.1.7: 5x5 – theoretical] (Averbakh and Berman, 1994)
- non-linear programming [2.1: 14x14] (Stowers and Palekar, 1993).

Since both location and routing problems are NP-hard under most scenarios, accordingly, their combination leads undoubtedly to a NP-hard problem, so big instances can hardly be solved by exact approaches (exception to this being the round-trip location problem [1.1.1]). Moreover, although it is possible to devise exact solution methods for many hard combinatorial problems, they may be too slow and inappropriate for practical purposes (mainly because hardly all variables are inserted into the model).

In this context, approximate or heuristic approaches are often more suitable than exact algorithms to achieve real-world needs. This is mainly due to their ability to provide high quality solutions, in reasonable time, to problems with significant computational complexity. These reasons have prompted researchers to increasingly use heuristics (see Appendix A), since they also have the advantage of producing more than one solution and are easier to understand, modify and implement.

LRP heuristic approaches are often classified according to the adopted framework (Nagy and Salhi, 2007). Here, a second classification based on the methods used to obtain the solution is proposed. Any LRP heuristic can therefore be classified using this two categories: the heuristic framework (Nagy and Salhi, 2007) and the heuristic methods. A single heuristic approach typically adopts a combination of the two: the framework, which defines how the location and routing phases interact and how information is passed on between them; and the methods, used in order to obtain the solution (possibly a combination of more than one).

The heuristic framework can be:

- Sequential – these can be "locate-first, route-second" or "route-first, locate-second". In both cases the problems are solved separately and sequentially, basically differing between them which problem is tackled first (Gerdessen, 1996; Murty and Djang, 1999). Most authors do not consider these as true LRP, due to the absence of an integrated view. Still, they are typically included in order to provide literature for benchmarking and solution quality evaluation. Some authors state that this approach may lead to the suboptimal design of logistics systems (Balakrishnan et al., 1987).
- Clustering-based – the routing phase is decomposed into the clustering of customers (one per potential depot or per vehicle route) and the actual route construction (through a VRP or a TSP). Afterwards, it can locate the depot in each cluster and find the corresponding route (Chan et al., 2001) or, design the route and then locate the corresponding depot (Barreto et al., 2007).
- Iterative – iteration is made between the location and routing phases. The problem is decomposed into the two subproblems, which are solved iteratively until some stopping

criteria are met (with feedback between both phases) (Salhi and Nagy, 2009; Schwardt and Fischer, 2009).

- Hierarchical (or nested, Nagy and Salhi, 1996a) – decisions are made at different levels. The main goal is the location decision while routing algorithms or route length estimation is used to obtain the associated cost. Hence, location is considered hierarchically superior (dominating problem) to the routing phase (Albareda-Sambola et al., 2007; Schittekat and Sörensen, 2009). Moreover, bilevel programming approaches (Marinakis and Marinaki, 2008a) can also be considered a hierarchical framework.

The following heuristic methods can be found:

- Tour construction and improvement – these methods are used for route construction/improvement, typically later combined with "add", "drop", or "swap" moves for depot location (where depots are respectively added, dropped, or swapped to/from the solution). Methods found in the literature are the Clarke and Wright (or "savings") algorithm (Tuzun and Burke, 1999; Chan et al., 2001) and k-opt heuristics (Branco and Coelho, 1990; Barreto et al., 2007).
- Lagrangian relaxation – in these methods, hard constraints are moved into the objective function in order to penalize it if they are not satisfied (Shen and Qi, 2007; Aksen and Altinkemer, 2008).
- Metaheuristic – these are present in many of the most recent publications in the area and currently include genetic algorithms (Su, 1998; Marinakis and Marinaki, 2008a), simulated annealing (Lin et al., 2002; Wu et al., 2002), tabu search (Lin and Kwok, 2006; Caballero et al., 2007), greedy randomized adaptive search procedure (Prins et al., 2006; Duhamel et al., 2010), particle swarm optimization (Marinakis and Marinaki, 2008b), and variable neighbourhood search (Melechovský et al., 2005).
- Neural network – methods which aim to simulate the human brain processing (Schwardt and Dethloff, 2005; Schwardt and Fischer, 2009).
- Fuzzy programming – fuzzy logic derives from the fuzzy set theory, where it is possible to have intermediate values between the usual boolean values. This applies when there is a significant degree of uncertainty regarding a situation. So far, the only paper addressing this type of methods is the work by Ehrgott and Verma (2001) on the transportation-location problem [2.1].

Objectives

The final tier of the second level of the taxonomy regards the objective function(s). The different papers are classified according to single- or multi-objective. The former encompasses a single objective (typically cost minimization) while the latter incorporates articles that study several objectives regarding location-routing.

Within single-objective models, the majority of papers fall upon the minimization of a linear combination of costs. These can be regarding: depot installation and operating, routes design, and vehicle fixed costs. Typically these costs are weighted and scaled down so that they relate to the same time horizon, in order to obtain a single objective.

Exceptions to cost minimization include:

- The round-trip location problem [1.1.1], where it is intended to minimize the maximum distance per route, which could be seen more as an equity measure, rather than cost minimization. This happens due to the need not to surpass a given route length (or cost) bound. Regarding this LRP variant only Kolen (1985) addresses cost minimization (namely, number of new facilities to install) besides the minimax aforementioned objective.
- The paper by Watson-Gandy and Dohrn (1973) where instead of minimizing costs it is intended to maximize profits.
- The work of Averbakh and Berman (1995), where the goal is to minimize the total waiting time of all customers. On Averbakh and Berman (1994) the abovementioned objective is combined with a surrogate of cost minimization (minimization of total tour length).
- Nema and Gupta (1999) use a composite single objective function consisting of total cost and total risk in response to a hazardous waste management model.

Even though most papers in this taxonomy (109 out of 128 – around 85%) deal with single-objective models, most real-life decisions cannot be accurately modelled with a single objective function. This causes growing interest in techniques taking into account multiple objectives. In the literature, although there are some works reviewing the LRP, seldom multi-objective models and approaches have been given much attention. Typically this aspect is relegated to a lesser role compared to the most addressed algorithmic approach.

When dealing with multi-objective models, one of the objectives is typically cost minimization (as in single-objective models). Other objectives include: risk minimization, work time and load imbalance minimization, and so forth.

In the reviewed multi-objective papers, objectives can be classified according to:

- cost minimization
- environmental aspects
- equitable distribution.

As follows, these three general categories will be examined, and the articles addressing them will be identified.

Cost minimization (or some surrogate such as minimization of total travel distance or number of depots to install) is, as previously stated, a traditional objective in location-routing models. Most single-objective models intend to minimize some measure of cost. Given this, it is to be expected that a significant amount of papers, in multi-objective, address cost minimization in some way. This is the case with almost all articles reviewed in this modelling approach, where some degree of cost is defined as an objective (the only exception being the work by Stowers and Palekar, 1993).

Nowadays, environmental aspects are critical in many real-world facility installation decisions. Such is the goal of this set of objectives: to minimize the risk to environment (and subsequently social rejection) caused by both the transportation and the proximity to a specific depot. Obviously, this is most applicable in scenarios where there is a degree of nuisance or hazard (or even possible risk in case of accident) relating to either the transported product or the normal functioning of the facility. Although, in most cases, this holds true, it is not always so. In this category is also included the satisfaction level, a measure used in the determination of the installation of desirable facilities, or as an effectiveness measure, representing the total demand serviced.

Given the probable social rejection in the transportation and location of undesirable products/facilities, there is a tendency to distribute equitably this obnoxious factor through the several demand points (usually population centres) in order to counter the typical opposition. This objective is addressed in several real-life situations and, as such, it has been incorporated into several models in the literature. Other equity objectives considered are regarding the balance of the work time and load of the vehicles, in an effort to make a correct distribution through all. These latter objectives are used only once, in the work by Lin and Kwok (2006). A final equity objective is the minimization of total distance to unmet demand. This objective incorporates the DM's desire to ensure that, in the future, the unmet demand can be met as easily as possible, by installing a new depot, or increasing the capacity of an existing one. This may be the case when the demand surpasses the (desired to install) supply, as in Johnson et al. (2002).

Table 2.3 provides a summary of the different main objectives considered so far in multi-objective LRPs. On Table 2.4 the articles classification of the multi-objective LRPs in the literature is made according to their objectives.

Table 2.3 Summary of the objectives addressed in articles using location-routing multi-objective models.

Objective type		Addressed objective
Cost minimization	C1	Minimization of the number of depots
	C2	Minimization of depot installation cost
	C3	Minimization of transportation cost
	C4	Minimization of travel distance
	C5	Minimization of travel time
	C6	Minimization of transportation burden (weight per distance)
	C7	Minimization of distance travelled by clients accessing depots
		Maximization of proximity to depots
	C8	Minimization of total costs (depot installation and transportation)
Environmental aspects	E1	Minimization of transportation risk (or nuisance)
	E2	Minimization of location risk (proximity to obnoxious depot)
	E3	Minimization of total risk (transportation and location)
	E4	Maximization of population satisfaction level (population serviced)
Equitable distribution	D1	Minimization of maximum transportation risk
	D2	Minimization of maximum location risk
	D3	Minimization of maximum total risk (transportation and location)
	D4	Minimization of work time imbalance
	D5	Minimization of load imbalance
	D6	Minimization of total distance to unmet clients demand
Others	O	Other objectives regarding a specific model

Table 2.4 LRP articles using multi-objective models.[a]

Article	Cost								Environment				Equity							
	C1	C2	C3	C4	C5	C6	C7	C8	E1	E2	E3	E4	D1	D2	D3	D4	D5	D6	O	Total
Ogryczak et al. (1989)		+	+																+	3
Zografos and Samara (1989)					+				+	+										3
List and Mirchandani (1991)								+			+				+					3
ReVelle et al. (1991)						+			+											2
Ogryczak et al. (1992)		+		+			+					+								4
Boffey and Karkazis (1993)				+					+		+									3
Stowers and Palekar (1993)											+				+					2
Jacobs and Warmerdam (1994)								+			+									2
Current and Ratick (1995)								+	+	+			+	+						5
Wyman and Kuby (1995)								+			+				+					3
Kulcar (1996)	+		+																	2
Giannikos (1998)								+	+				+	+						4
Murty and Djang (1999)	+			+			+													3
Ehrgott and Verma (2001)				++																2
Johnson et al. (2002)								+				+						+		3
Lin and Kwok (2006)								+								+	+			3
Alumur and Kara (2007)								+	+											2
Caballero et al. (2007)		+	+						+	+			+							5
Boffey et al. (2008)		+	+						+					+						4

[a] For further explanation on the abbreviations, the reader is referred to Table 2.4.

It should be noted that out of the 19 multi-objective papers addressed, 15 are transportation-location problems [2.1], from which most are regarding the location of undesirable facilities (and/or transportation of HAZMATs). This type of problems is inherently multi-objective in that the trade-offs between cost and exposure risk or nuisance to communities must be simultaneously considered. This will be further addressed in Chapter 4.

2.3 Summary

This chapter made an introduction to and an overview of the models handling both location and routing. Also, a review of the main issues regarding LRPs (the integrated approach) was made and a taxonomy proposed, in order to identify existing gaps and foster future studies.

Among the most overlooked LRP variants in the literature, one can distinguish between problems which have recently emerged (usually as the result of a recent real-world application) and problems which have attracted little attention over the years. Within the former there is the planar LRP [1.1.5], the plant-cycle location problem [1.1.6], and the many-to-many LRP [2.2]. The latter encompass the LARP [1.1.3], the Hamiltonian p-median problem [1.1.4], the stochastic LRP [1.2.2], the dynamic LRP [1.2.3], and the vehicle routing-allocation problem [2.3]. All of the above LRP variants have been mostly addressed using heuristic approaches and single-objective models.

Besides the mentioned taxonomy, current approaches and objectives on location-routing were also analysed in this chapter.

Due to the lack of comprehensive studies benchmarking solution quality, no absolute conclusion can be made on the effectiveness of a single method or approach. Nevertheless, on exact approaches, branch-and-cut and branch-and-price methods have been able to solve instances of up to, respectively, 200 clients and 50 potential depot locations for the LARP [1.1.3] (Ghiani and Laporte, 1999), and 100 clients and 10 potential depot locations for the general LRP [1.1.7] (Berger et al., 2007). On the other hand, dynamic programming seems unfit for this type of problems as it was only solved (theoretically) once in a very small tree network (Averbakh and Berman, 1994).

Regarding heuristic approaches (possibly more prone to be used in real-world location-routing scenarios), the lack of strong lower bounds makes difficult to draw conclusions on the performance of heuristics for large instances. Albeit some authors (e.g. Nagy and Salhi, 2007, and Schittekat and Sörensen, 2009) point hierarchical frameworks as being more likely to obtain better results, both clustering-based and iterative frameworks have proven to be equally competitive (as is the case for the CLRP [1.1.2], which has recently been receiving the most attention). Moreover, at this point, the largest test instances solved (using heuristic approaches and excluding case-oriented papers) include 400 clients and 400 potential depot locations. Being mostly randomly generated, they may still lack the scalability to be used when analysing real-world scenarios.

Concerning objectives, a more thorough analysis was presented for multi-objective models. In spite of the fact that in real-life most facility installation situations have to consider several opposing objectives, this category has been surprisingly frequently overlooked in the literature. However, this approach is essential to correctly tackle the location(-routing) of (semi-obnoxious) facilities which is one of the main subjects of this thesis.

Chapter 3

Basic Location-Routing Problems

As shown in the previous chapter there are various location-routing models. Moreover, according to Nagy and Salhi (2007), about a fifth of the literature address real applications, providing evidence of the importance and usefulness of these models.

However, location(-routing) decisions are application oriented, meaning, their structural form (objectives, constraints, and variables) is determined by the specific problem under study. Consequently, it does not exist a general model appropriate for all potential or existing applications (Current et al., 2002). This justifies an effort to formulate new models and to adapt existing ones, as well as to devise efficient solution techniques to essential models. These studies can later be adapted to address specific applications.

The purpose of this chapter is to formally define, review, and develop new approaches to two specific problems in the location-routing problem (LRP) literature, which can be seen as some of its most basic models: the capacitated LRP (CLRP) and the location-arc routing problem (LARP). These problems derive, respectively, from the capacitated vehicle routing problem (CVRP) (Laporte et al., 1986) and the arc routing counterpart, the capacitated arc routing problem (CARP) (Ghiani and Laporte, 2001).

The approaches developed for both problems were implemented and results obtained. The collected data were analysed using exploratory data analysis (EDA) (Hoaglin et al., 2000), parametric (Box et al., 2005) and non-parametric tests (Kvam and Vidakovic, 2007), and multivariate data analysis (Hair et al., 2009).

These new approaches may allow to, in the future, address more closely similar real-world applications.

3.1 Capacitated Location-Routing Problem

The CLRP derives from the CVRP and is one of the most addressed problems in the LRP literature (as seen in Chapter 2). In this section, the CLRP is formally defined, methods in the literature are reviewed, and a newly developed heuristic approach is presented (Lopes et al., 2009).

The approach is a metaheuristic based on guided local search embedded in a hybrid extended savings algorithm, as well as used to control a reduced composite local search. It uses some concepts of the AGES metaheuristic (Mester and Bräysy, 2005) applied favourably to the vehicle-routing problem and will be called active guided search (AGS).

The proposed AGS metaheuristic, after obtaining a cooperative starting solution (with both location and routing taken into consideration), proceeds to an intensification phase. In the latter, a facility location algorithm is performed to obtain the best locations. Then it attempts to find the best route configuration for the given locations. This is done iteratively until a stopping criterion is met and can be seen as a hierarchical framework. Results for the AGS are presented and compared with other approaches in the literature using benchmark instances.

3.1.1 Problem Definition

In the CLRP only two levels are considered (clients and depots) and the only route constraints are regarding the vehicle capacity (a fleet of identical vehicles with homogeneous capacity of a single product is assumed). Capacity constraints may also be assigned to each depot. This problem can be seen as an extension to the CVRP.

The CLRP can be formally defined on a weighted and directed graph $G = (V, A)$ where V is a vertex set and A is a set of arcs. V comprises a subset J of m potential facility locations and a subset $I = V \setminus J$ of n clients. The traversal non-negative cost of any arc $a = (i, j)$ in the arc set A is given by c_a (or c_{ij} between vertex i and j). Each client $i \in I$ has a given demand d_i, serviced once, and is to be assigned to a single facility $j \in J$ with capacity w_j. The shipment of clients demand is carried out by a set of K vehicles, with homogeneous capacity Q, which return to the departure depot at the end of the route.

There is a fixed cost f_j incurred when opening a depot at each potential site, and a distribution cost associated with routing which includes the cost of traversed arcs (total travelled distance), and the fixed cost F of acquiring a vehicle.

The objective is to determine the set of depots to open and the tracing of the routes departing from each open depot, in order to minimize a total cost encompassing the fixed cost of opening depots and the total distribution cost.

Being S any subset of vertices ($S \subset V$), $\delta^+(S)$ ($\delta^-(S)$) is the set of arcs leaving (entering) S, and $L(S)$ is the set of arcs containing both extremities in S. If S contains a single vertex v, $\delta^+(v)$ is a simplification for $\delta^+(\{v\})$. The following binary variables are used: x_{ak}, equal to one if arc a is used in the route performed by vehicle $k \in K$; y_j, equal to one if depot j is to be opened; and y_{ij}, equal to one if client i is assigned to depot j. The CLRP is then formulated as (Prins et al., 2006):

$$(\text{CLRP}_1) \quad \min \quad Z^1 = \sum_{j \in J} f_j y_j + \sum_{a \in A} \sum_{k \in K} c_a x_{ak} + \sum_{k \in K} \sum_{a \in \delta^+(J)} F x_{ak} \tag{3.1}$$

$$\text{s.t.:} \quad \sum_{k \in K} \sum_{a \in \delta^-(i)} x_{ak} = 1 \qquad \forall i \in I, \tag{3.2}$$

$$\sum_{i \in I} \sum_{a \in \delta^-(i)} d_i x_{ak} \leq Q \qquad \forall k \in K, \tag{3.3}$$

$$\sum_{a \in \delta^+(i)} x_{ak} - \sum_{a \in \delta^-(i)} x_{ak} = 0 \qquad \forall i \in V, \forall k \in K, \tag{3.4}$$

$$\sum\nolimits_{a \in \delta^+(J)} x_{ak} \leq 1 \qquad \forall k \in K, \qquad (3.5)$$

$$\sum\nolimits_{a \in L(S)} x_{ak} \leq |S| - 1 \qquad \forall k \in K, \forall S \subseteq I, \qquad (3.6)$$

$$\sum\nolimits_{a \in \delta^+(j) \cap \delta^-(I)} x_{ak} + \sum\nolimits_{a \in \delta^-(i)} x_{ak} \leq 1 + y_{ij} \qquad \forall i \in I, \forall j \in J, \forall k \in K, \qquad (3.7)$$

$$\sum\nolimits_{i \in I} d_i y_{ij} \leq w_j y_j \qquad \forall j \in J, \qquad (3.8)$$

$$x_{ak} \in \{0,1\} \qquad \forall a \in A, \forall k \in K, \qquad (3.9)$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \qquad (3.10)$$

$$y_{ij} \in \{0,1\} \qquad \forall i \in I, \forall j \in J. \qquad (3.11)$$

The objective function (3.1) seeks to minimize the abovementioned costs. Constraints (3.2) are the set partitioning constraints which require that each client i be serviced exactly once by a single route. Constraints (3.3) require vehicle capacity to be obeyed. Constraints (3.4) and (3.5) ensure the continuity of each route (flow conservation constraints) and a return to the departure depot, while constraints (3.6) are the well-known subtour elimination constraints. Constraints (3.7) ensure that a client can only be assigned to a depot if there is a route linking them. Depot capacity constraints are satisfied thanks to inequalities (3.8). Finally, (3.9), (3.10), and (3.11) are standard binary constraints.

3.1.2 Recent Algorithmic Developments

As the CLRP results from the combination of two NP-hard problems, the facility location problem and the CVRP, it belongs to the category of NP-hard problems. As such, it is very difficult to solve it using exact algorithms, the only being the one presented in the work by Laporte et al. (1986). Due to this difficulty, several heuristic approaches have appeared in the literature which will now be detailed.

The CLRP has emerged as one of the most addressed LRP. A two-phase tabu search architecture for CLRP with uncapacitated depots was developed by Tuzun and Burke (1999). The heuristic uses a hierarchical framework (for each feasible location, a simple routing phase is performed) evaluating neighbouring moves and subsequently adding a depot until the total cost is increased.

Wu et al. (2002) studied a CLRP with both homogeneous and heterogeneous fleets and a limited number of vehicles. These authors present an iterative procedure that relies on a simulated annealing algorithm combined with a tabu list.

In Barreto et al. (2007) a clustering based heuristic is presented for tackling the CLRP with no vehicle acquisition cost. Several clustering methods are used to obtain the routing data and then a facility location problem is solved with the collapsed routes. Marinakis and Marinaki (2008a, 2008b) solve the same problem using, in the two works, respectively, a bilevel genetic algorithm and hybrid particle swarm optimization.

Prins et al. (2006) develop an extended savings heuristics, which is used in a greedy randomized adaptive search procedure (GRASP). Finally, Prins et al. (2007) present a more effective heuristic where facility location (through Lagrangian relaxation) and vehicle routing (using a granular tabu search) is performed iteratively. These two works consider the same CLRP as defined previously and to be addressed henceforth (in the AGS).

3.1.3 Active Guided Search Metaheuristic

The AGS metaheuristic follows some of the ideas developed by Mester and Bräysy (2005) for the vehicle routing problem (VRP), and is composed of two phases. The first phase aims at providing a starting solution for the second phase. The starting solution is obtained by hybridizing an extended savings method (Prins et al., 2006) which adapts the well known “savings” algorithm (Clarke and Wright, 1964) to the LRP.

The second phase can be seen as an intensification phase, where an attempt is made to improve the starting solution. This second stage is composed of a two-step procedure. Firstly, a more thorough local search is performed using the GLS metaheuristic (Voudouris, 1997). GLS operates by increasing the problems objective function with a penalty term based on a specific solution feature to be avoided in a near-optimal solution. In this method, the facility location problem heuristic (Filho and Galvão, 1998) handles the location of facilities, hence the penalized feature will be regarding the routing component of the solution (long arcs). The GLS is used to guide a reduced composite local search procedure consisting of a relocate algorithm (Savelsbergh, 1992) and a 2-opt algorithm (Prins et al., 2006) (both used in intra- and inter-route improvements). When no more improvements can be found for a number of iterations the algorithm proceeds to the second step. It then performs a second run of the hybrid extended savings algorithm, only considering the open depots obtained in the best found solution of the former step.

Hybrid Extended Savings Heuristic

The “savings” or Clarke and Wright algorithm (Clarke and Wright, 1964) is a well known constructive algorithm often used in the VRP. The algorithm starts by assigning each client to a single depot (each client is supplied by a dedicated route). Then, each pair of routes whose total load does not exceed the maximum vehicle capacity is inspected to evaluate the saving (the gain obtained from merging both routes, with four possible combinations per pair of routes). The merger providing the largest positive saving is performed continuously until no feasible merge can be found.

Prins et al. (2006) developed an extended version of the abovementioned heuristic for the CLRP where, similarly, the solution starts by building the flower-like trivial solution, closing all depots with no clients assigned. Then, each route merge is evaluated considering a possible reassignment to all depots (Figure 3.1), hence resulting in $4m$ possible merges for each pair of routes (R, S). The saving σ is computed as follows:

$$\sigma = F + c_{ri} + c_{jr} + c_{sk} + c_{ls} - c_{jk} - c_{ti} - c_{lt} + f_r\theta_r + f_s\theta_s - f_t(1 - y_t). \tag{3.12}$$

Where: r, s, and t are respectively the depot of route R, route S, and the depot currently considered to be assigned to the merged route; i, j, k, and l are the clients connected to the depots in each route; θ_r (θ_s) is a binary value equal to 1 if depot r (s) supplies no more routes after the merger (thus can be closed); and y_t is a binary value (defined earlier for the formulation) equal to 1 if depot t is already open before the merge.

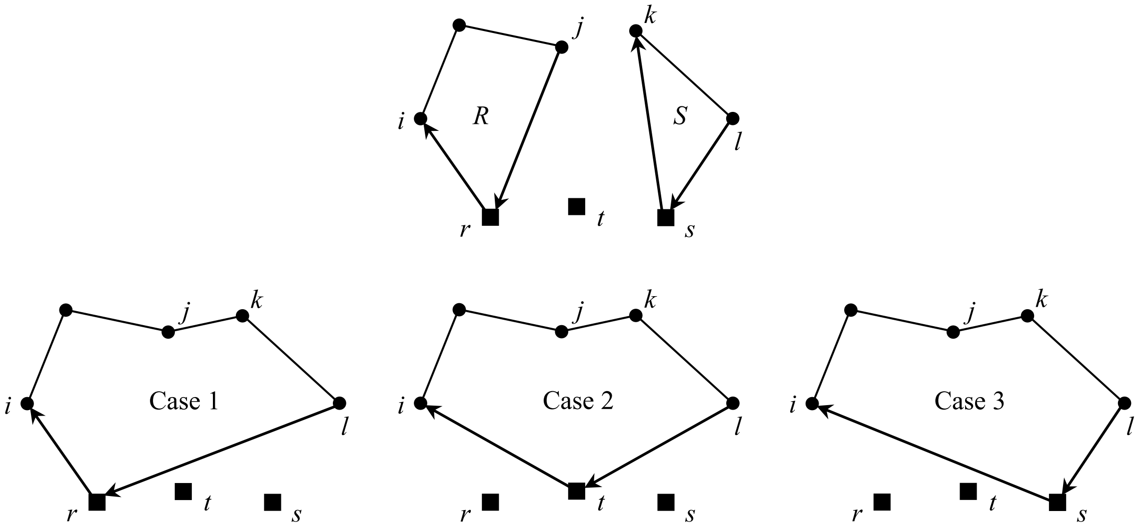


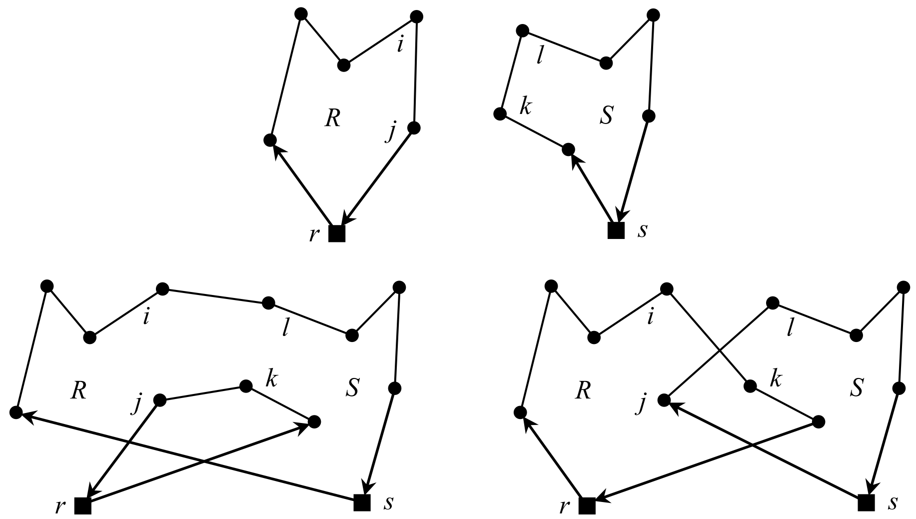
Figure 3.1 Some merges in the extended savings algorithm (Prins et al., 2006).

It has been shown (e.g. Russell, 1995) that hybridizing the building of a solution often leads to better results. Hence, in AGS, this method was adapted and hybridized to provide a "good" starting solution. Firstly, rather than making a direct assignment to the clients in the first step, a facility location problem is solved using a tabu search (TS) heuristic (Filho and Galvão, 1998). Also, rather than randomly choosing the saving out of a restricted candidate list, the merger that provides the largest feasible saving is always performed. Finally, each time the number of route merges performed ($merges$) matches 20% of the total number of clients (n), an improvement is made using a composite local search and a GLS to guide a reduced composite local search.

Composite Local Search. The composite local search is composed of a facility location algorithm (Filho and Galvão, 1998) applied only once, a relocate algorithm (Savelsbergh, 1992) and a 2-opt algorithm (Prins et al., 2006), the latter two performing intra- and inter-route moves applied sequentially until no improvement can be found.

The facility location algorithm chosen was the TS developed by Filho and Galvão (1998) due to its reduced CPU time and near-optimal results. However, since at a given point the solution has routes instead of individual clients, in order to apply the heuristic, each route is collapsed into a single client and the considered distance is the smallest insertion cost of the depot in the original route, similarly to Barreto et al. (2007).

The relocate algorithm performs in the same way as the one used in VRPs (Savelsbergh, 1992), where a single client is reinserted in another position inside the current route or in another route provided there is an improvement to the solution. In the latter case, an adaptation was made for the CLRP in order to account for the depot capacity constraints. Finally, the 2-opt algorithm (Prins et al., 2006) inside the routes is equivalent to the well known 2-opt move (Lin and Kernighan, 1973) whereas the moves between different routes have to consider depot capacity constraints and depot reassignment (Figure 3.2). This 2-opt procedure implements the first found improvement rather than the best (empirically found to be better by Hansen and Mladenović, 2006). It should be noted that a similar 3-opt algorithm (Branco and Coelho, 1990) was implemented but, since an increased computation time was needed and no significant improvement to the final solutions was obtained, it was removed from the proposed AGS metaheuristic.



(i,j) and (k,l) replaced by (i,l) and (j,k) (i,j) and (k,l) replaced by (i,k) and (j,l)

To start and end at the same depot, the new trips are reassigned to r and s

Figure 3.2 Example of the 2-opt procedure performing inter-route moves (Prins et al., 2006).

Guided Local Search. Guided local search (GLS) is performed in order to escape local optima. GLS (Voudouris, 1997) is a memory-based metaheuristic shown to be effective for several combinatorial optimization problems (Tsang and Voudouris, 1997; Voudouris and Tsang, 1999; Mills et al., 2003; Mester and Bräysy, 2005). The metaheuristic main concept is to penalize a solution feature which is considered not to exist in a near-optimal solution. In the CLRP, as the facility location algorithm handles the location, the undesirable solution feature is regarding the routing, hence long arcs are penalized.

A single arc is penalized every time the reduced composite local search (which differs from the complete version by not performing the facility location algorithm) gets stuck in a local minimum.

In order to choose the arc, the "utility" function in equation (3.13) is used:

$$U = \frac{c_{ij}}{1 + p_{ij}} \tag{3.13}$$

where p_{ij} is the number of times the arc (i, j) has been penalized. The arc with the highest utility value is chosen to be penalized as follows:

$$p_{ij}^{'} = p_{ij} + 1. \tag{3.14}$$

The purpose is to penalize the longer arcs (subsequently with higher cost) although the utility of doing so decreases as the feature is increasingly penalized. Then, a new cost $c_{ij}^{'}$ is obtained for the reduced composite local search:

$$c_{ij}^{'} = c_{ij} + p_{ij}\lambda L \tag{3.15}$$

where λ is a parameter (with value empirically found: $\lambda = 0.05$) and L is the average length of the arcs in the current solution. After the new cost is calculated, the reduced composite local search is performed on a neighbourhood restricted to the set of ε closest routes. This is called a penalty variable neighbourhood (PVN) which size is adjusted dynamically during the search. In order to obtain the PVN, firstly, the routes are sorted according to the proximity to the route of the penalized arc. Then, the following criteria are adopted for selecting the minimum number of routes in the PVN (ε_{min}): (1) the route the chosen arc is in; (2) all routes which closest distance to the selected arcs route is zero (e.g. routes sharing the same depot); and (3) the two following closest routes. The maximum size of the PVN (ε_{max}) is set to the maximum between ε_{min} and 75% of the number of routes in the current solution. Thus, the size of the PVN (ε) can be obtained by:

$$\varepsilon = \varepsilon_{min} + \lceil(\varepsilon_{max} - \varepsilon_{min})rnd\rceil \tag{3.16}$$

with rnd as a random number between 0 and 1 obtained at each iteration.

This proposed GLS is performed until a given number β of iterations is attained without improvements to the solution. Once this number of iterations is achieved, the proposed implementation resets the penalties and restarts for an ω number of times, using the best found solution after performing on it the complete composite local search. Embedded in the hybrid extended savings, the following parameters were used: $\beta = 250$ and $\omega = 5$. These were chosen after a preliminary analysis that showed: for β, most improvements were obtained within the next 250 iterations after the last improvement and hardly any after 500 iterations; for ω, after five restarts rarely new best solutions were found.

Intensification

After obtaining the starting solution from the hybrid extended savings algorithm, the metaheuristic proceeds to attempt to improve it with a two step procedure. This can be seen as an intensification phase.

The first step of this phase is again based on the implemented GLS which, in this case, is performed with a more thorough search ($\beta = 500$). The second step is intended to further intensify the route optimization, thus a second run of the initial algorithm (hybrid extended savings heuristic) is performed. In this case, all the depots which were closed in the last best solution are removed from the problem, with the exception of one randomly chosen. This provides a small diversification and also insures that depot capacity constraints are not violated (in cases where, for the facility location problem, the open depots available capacity is highly restrictive).

After performing the second step, the algorithm returns to the first, where the solution is once more explored for better results (now with all depots). In order to improve the accuracy of the AGS metaheuristic this intensification phase can be done iteratively until stopping criteria are met (e.g. no better solution can be found in both steps). However, in the results presented in this work, the second phase is performed only once.

A flowchart with the main steps of the proposed AGS metaheuristic can be seen in Figure 3.3.

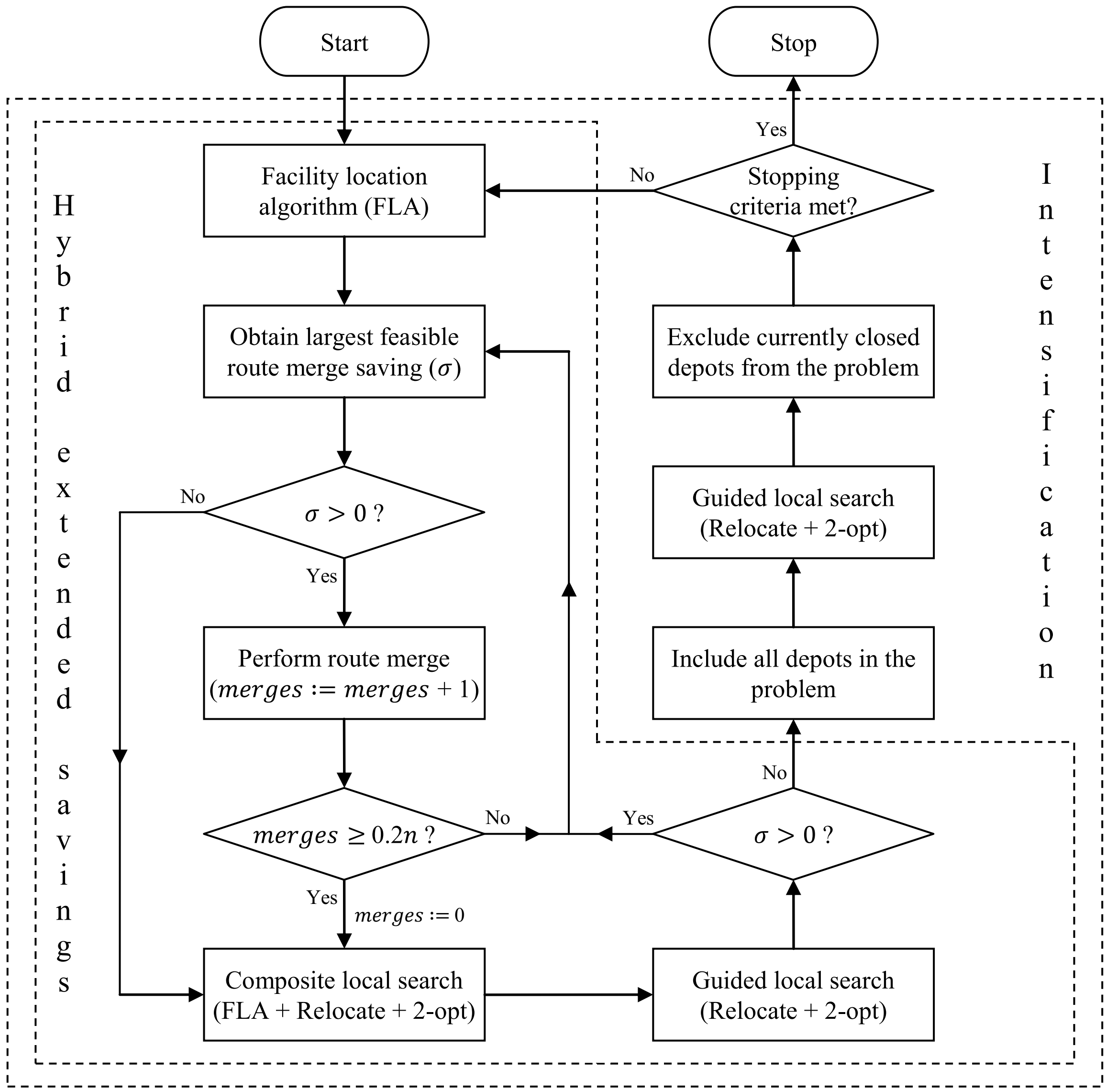


Figure 3.3 Flowchart of the AGS metaheuristic for the CLRP.

3.1.4 Computational Results

In this section experimental results obtained to ascertain the performance of the proposed metaheuristic are presented. Implementation issues and the used data sets are briefly described. Finally, a comparative analysis of the suggested AGS metaheuristic with other recent methods for the CLRP is made.

Implementation and Benchmark Instances

The metaheuristic was coded in C# and the results obtained using a 3.00 GHz Intel Xeon E5450 Quad Core CPU with 8 GB of RAM and Windows XP (no parallel processing was used).

In order to assess the efficiency of the proposed method, three sets of benchmark instances for the CLRP were used. These are the instances by Tuzun and Burke (1999), Barreto et al. (2007), and Prins et al. (2006). The sets have, respectively, 36, 19, and 28 problem instances, consider complete undirected graphs and Euclidean distances.

In the set of randomly generated instances by Tuzun and Burke, the number of clients is $n \in \{100, 150, 200\}$ and the number m of depots is 10 and 20. The vehicle capacity $Q = 150$, vehicle fixed cost $F = 10$, and the clients have demand following a uniform distribution in the range [1, 20]. Spatial distribution of the clients and depots was controlled using the number cl of clusters (0, 3, and 5, where 0 refers to uniformly distributed clients) and the percentage of clients belonging to a cluster ($clratio$, equal to 75% or 100%).

The benchmark instances proposed by Barreto et al. are available from http://sweet.ua.pt/~iscf143/ and were either obtained from the literature or adapted from VRP instances. The number of clients n ranges from 12 to 318, the number of depots m from 2 to 15, and several different values of vehicle capacity were considered. Moreover, there is no vehicle fixed cost.

The last set of instances by Prins et al. was randomly generated (available at http://prodhonc.free.fr/Instances/instances_us.htm) with $n \in \{20, 50, 100, 200\}$, $m \in \{5, 10\}$, $Q \in \{70, 150\}$, and $F = 1000$. Clients are spatially distributed in clusters ($cl \in \{0, 2, 3\}$ with 0 meaning a uniform distribution) and distances are multiplied by 100 and rounded up to the next integer.

Comparative Analysis

Results for the AGS metaheuristic were obtained for the three sets of benchmark instances and compared with results in the literature. Twenty runs were performed for each instance from which was found the average and best result (and corresponding computing time).

Results for the benchmark instances can be seen in Tables 3.1, 3.2 and 3.3. The first columns of the tables display the name of the instances while the following three show, respectively, the number of clients (n), the number of depots (m), and the vehicles capacity (Q). Then, follows the data regarding the generation of the instances: the number cl of clusters (for the first and third sets) and the clustering ratio $clratio$ (for the first set). The AGS average results (Avg.), best results (Best) and CPU times in seconds (corresponding to the best results) are reported afterwards.

In the first and third sets (Table 3.1 and Table 3.3) best results are compared to the best known results (BKR) in the literature, by presenting Gap_{BKR} which shows, in percentage, the gap between the obtained best and the currently best known results. On the instances by Barreto et al. (Table 3.2) best results are compared to the currently known best lower bound (LB). Hence, Gap_{LB} refers to the gap (in percentage) to the lower bound.

Whenever a value with an asterisk is found it means the cost is the optimal value of that specific instance. Underlined values (9, 8, and 1 respectively for the first, second, and third sets of instances) point to newly found best results obtained by the proposed method from which one is an optimal solution (instance Gaskell67-32x5a, see Figure 3.4). Moreover, AGS is able to match 14 previously known best solutions from which 10 are optimal (out of the 12 previously known).

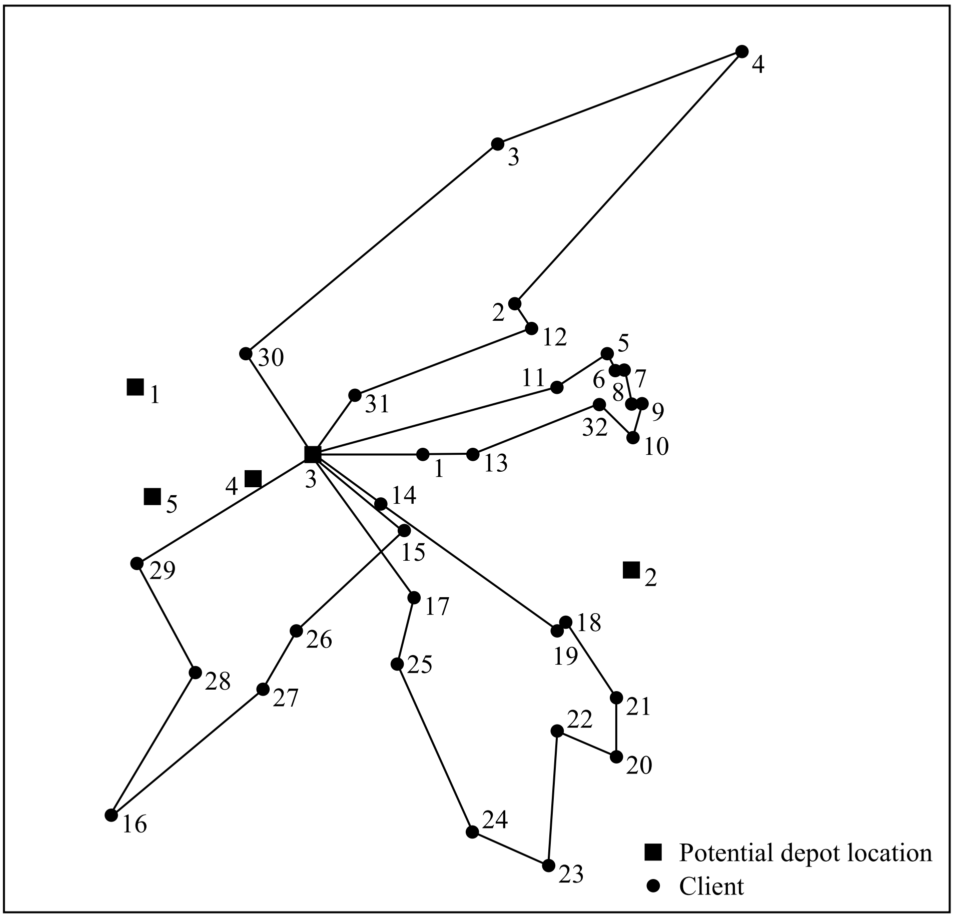


Figure 3.4 Graphical representation of the newly found optimal solution obtained by the AGS metaheuristic for instance Gaskell67-32x5a.

The empirical distribution of CPU time, Gap_{BKR}, and Gap_{LB} showed heavily skewed batches and/or outlying data points. Thus, as the median is a resistant statistic, whereas the sample mean (average) is not, Tables 3.1, 3.2, and 3.3 also include the corresponding median values. The purpose of the average values is the comparison with algorithms from the literature (Table 3.4), as these are the values usually reported.

Table 3.1 Results for the instances by Tuzun and Burke (1999).

							AGS			
Instance	n	m	Q	cl	$clratio$	BKR	Avg.	Best	CPU	Gap_{BKR}
1 111112	100	10	150	0	1	1468.40	1509.17	1469.43	30.4	0.07
2 111122	100	20	150	0	1	1449.20	1485.47	1449.20	39.8	0.00
3 111212	100	10	150	0	1	1396.46	1435.11	1399.69	46.0	0.23
4 111222	100	20	150	0	1	1432.29	1470.47	1432.29	38.7	0.00
5 112112	100	10	150	3	0.75	1167.53	1179.20	1167.53	39.5	0.00
6 112122	100	20	150	3	0.75	1102.70	1125.55	1103.53	83.3	0.08
7 112212	100	10	150	3	1	793.97	839.18	806.16	30.5	1.54
8 112222	100	20	150	3	1	728.30	732.34	728.40	42.6	0.01
9 113112	100	10	150	5	0.75	1238.49	1265.19	1238.70	35.7	0.02
10 113122	100	20	150	5	0.75	1246.34	1259.83	1247.56	38.1	0.10
11 113212	100	10	150	5	1	902.38	933.37	917.97	25.9	1.73
12 113222	100	20	150	5	1	1021.31	1046.25	1019.35	59.3	-0.19
13 131112	150	10	150	0	1	1866.75	1957.49	1930.33	71.6	3.41
14 131122	150	20	150	0	1	1841.86	1891.89	1847.33	87.4	0.30
15 131212	150	10	150	0	1	1981.37	2021.09	1976.66	162.1	-0.24
16 131222	150	20	150	0	1	1809.25	1861.19	1808.81	90.6	-0.02
17 132112	150	10	150	3	0.75	1448.27	1461.90	1422.27	73.4	-1.80
18 132122	150	20	150	3	0.75	1444.25	1466.85	1447.13	97.1	0.20
19 132212	150	10	150	3	1	1206.73	1216.13	1207.21	142.9	0.04
20 132222	150	20	150	3	1	931.94	955.03	936.01	227.7	0.44
21 133112	150	10	150	5	0.75	1699.92	1741.78	1712.03	186.3	0.71
22 133122	150	20	150	5	0.75	1401.82	1431.53	1402.37	83.6	0.04
23 133212	150	10	150	5	1	1199.51	1235.93	1211.48	593.2	1.00
24 133222	150	20	150	5	1	1152.86	1184.00	1158.18	88.7	0.46
25 121112	200	10	150	0	1	2281.78	2322.27	2252.34	168.0	-1.29
26 121122	200	20	150	0	1	2185.55	2259.89	2182.26	408.2	-0.15
27 121212	200	10	150	0	1	2234.78	2277.77	2224.05	118.5	-0.48
28 121222	200	20	150	0	1	2259.52	2294.03	2263.99	286.6	0.20
29 122112	200	10	150	3	0.75	2101.90	2141.04	2107.65	130.4	0.27
30 122122	200	20	150	3	0.75	1709.56	1735.73	1714.17	116.8	0.27
31 122212	200	10	150	3	1	1467.54	1522.43	1473.54	146.2	0.41
32 122222	200	20	150	3	1	1084.78	1177.25	1091.49	304.8	0.62
33 123112	200	10	150	5	0.75	1973.28	2002.02	1968.29	168.9	-0.25
34 123122	200	20	150	5	0.75	1957.23	1986.41	1950.15	179.5	-0.36
35 123212	200	10	150	5	1	1771.06	1811.41	1778.21	157.2	0.40
36 123222	200	20	150	5	1	1393.62	1535.41	1405.37	215.7	0.84
								Average	133.8	0.24
								Median	93.9	0.09

The empirical distribution of CPU time depicted in Figure 3.5 (concerning Table 3.1) shows the batch is heavily skewed (skewness = 2.18) and two outlying data points (moderate = 408.2; severe = 593.2). Median time is 93.85 seconds and 75% (Q3) CPU times spread to 168.45 seconds.

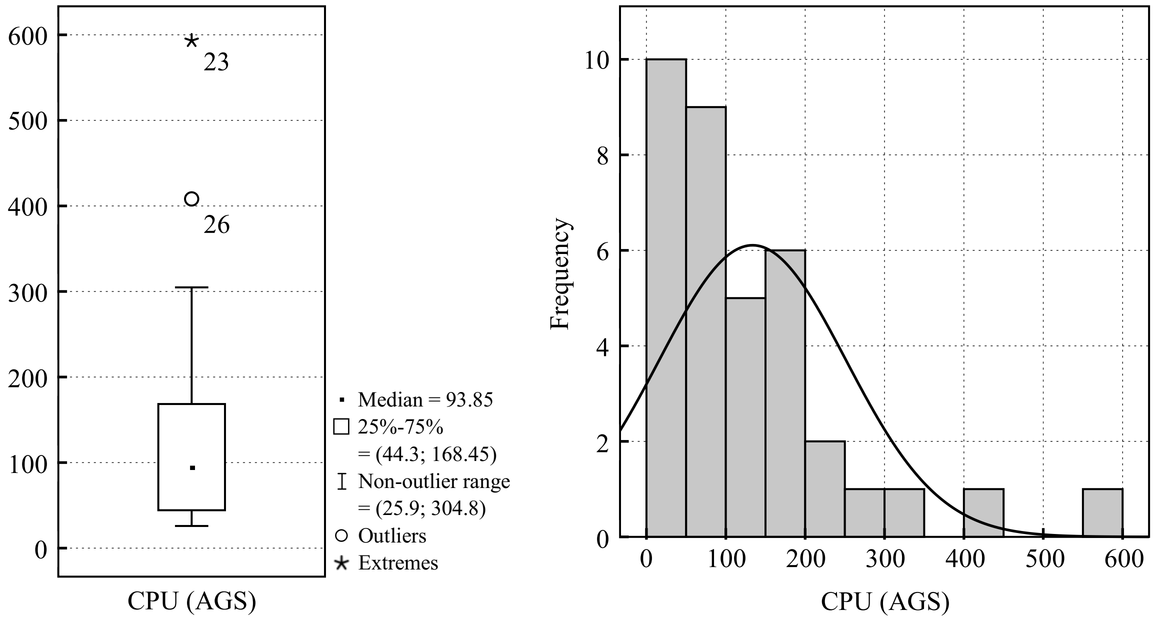


Figure 3.5 Boxplot (left) and histogram (right) for the CPU time in seconds (Table 3.1).

Figure 3.6 with severe (-1.80; 3.41) and moderate (-1.29; 1.54; 1.73) outliers suggests the use of the median value = 0.09% to characterize the location of Gap_{BKR} data (Table 3.1). Moreover, 75% of Gap_{BKR} values are less than 0.43%.

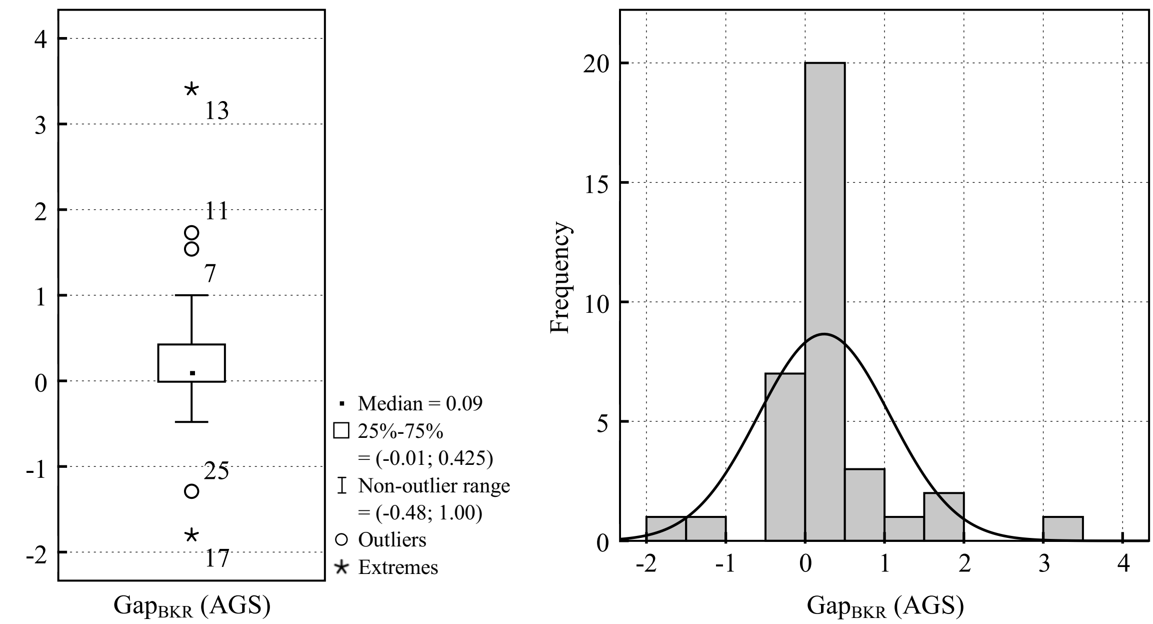


Figure 3.6 Boxplot (left) and histogram (right) for the Gap_{BKR} in percentage (Table 3.1).

Categorized boxplots in Figure 3.7 show the relationship among Gap_{BKR}, number of clients (n), and depots (m), regarding Table 3.1. Good results for Gap_{BKR} remain irrespectively of the increase of clients and/or depots.

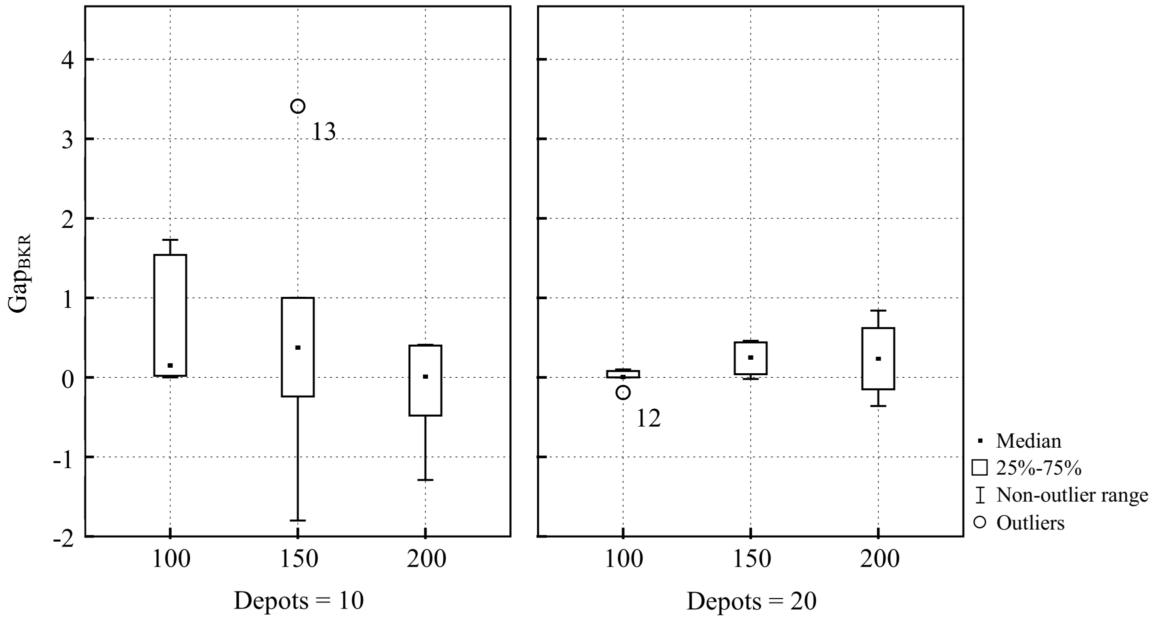


Figure 3.7 Categorized boxplots for the Gap_{BKR} in percentage: clients = 100, 150, and 200; depots = 10 and 20 (Table 3.1).

The abovementioned good results, for the AGS metaheuristic, were statistically confirmed using a dependent t-test for BKR and best AGS. Figure 3.8 shows the boxplots for both sets of values.

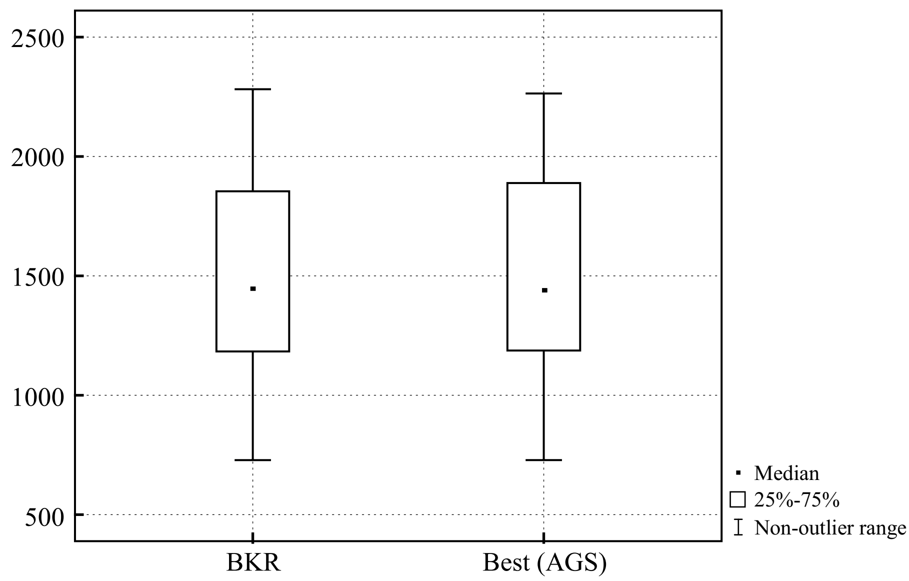


Figure 3.8 Boxplots for best known results (BKR) and best obtained values (best AGS) (Table 3.1).

The null hypothesis of equal means is not rejected for a significance level (α) of 5%, as the p-value = 0.24 and bigger than $\alpha = 0.05$.

Table 3.2 Results for the instances by Barreto et al. (2007).

	Instance	n	m	Q	LB	AGS Avg.	AGS Best	AGS CPU	Gap_{LB}
1	Christofides69-50x5	50	5	160	551.1	578.5	565.8	11.0	2.67
2	Christofides69-75x10	75	10	140	791.4	865.4	845.9	21.0	6.89
3	Christofides69-100x10	100	10	200	818.1	861.4	843.9	69.0	3.15
4	Daskin95-88x8	88	8	9000000	347.0	376.9	368.8	43.6	6.28
5	Daskin95-150x10	150	10	8000000	43406.0	45284.1	44510.0	62.2	2.54
6	Gaskell67-21x5	21	5	6000	*424.9	429.9	*424.9	3.5	0.00
7	Gaskell67-22x5	22	5	4500	*585.1	586.5	*585.1	1.8	0.00
8	Gaskell67-29x5	29	5	4500	*512.1	565.4	515.1	5.7	0.59
9	Gaskell67-32x5a	32	5	8000	*562.2	565.2	*562.2	5.1	0.00
10	Gaskell67-32x5b	32	5	11000	*504.3	508.8	*504.3	4.6	0.00
11	Gaskell67-36x5	36	5	250	*460.4	464.7	*460.4	4.4	0.00
12	Min92-27x5	27	5	2500	*3062.0	3062.3	*3062.0	2.4	0.00
13	Min92-134x8	134	8	850	5423.0	5925.4	5729.3	35.4	5.65
14	Or76-117x14	117	14	150	12048.4	12646.3	12422.0	54.8	3.10
15	Perl83-12x2	12	2	140	*204.0	215.0	*204.0	0.8	0.00
16	Perl83-55x15	55	15	120	1074.8	1122.4	1112.3	29.7	3.49
17	Perl83-85x7	85	7	160	1568.1	1635.0	1625.2	22.4	3.64
18	Perl83-318x4a	318	4	25000		582503.1	570200.9	772.3	
19	Perl83-318x4b	318	4	8000		692152.5	674591.9	443.2	
							Average	83.8	2.24
							Median	21.0	2.54

Figure 3.9 shows an asymmetric distribution (skewness = 3.13) as well as two severe outliers (443.2; 772.3) for the CPU time (Table 3.2). Median time is 21.0 seconds and 75% (Q3) CPU times are less than 54.9 seconds.

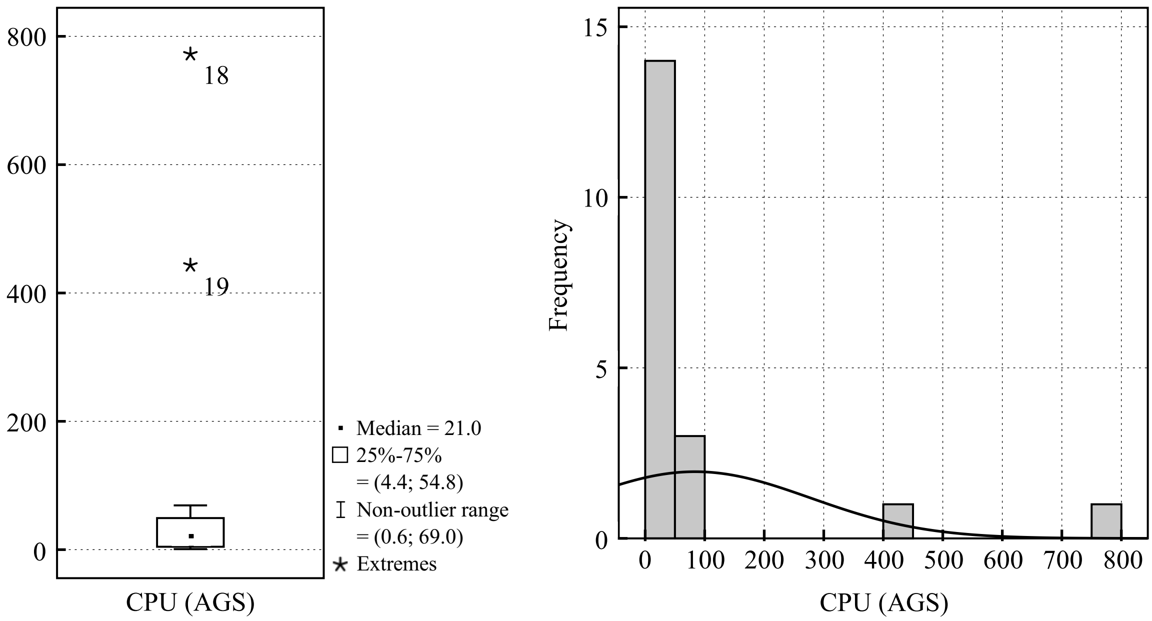


Figure 3.9 Boxplot (left) and histogram (right) for the CPU time in seconds (Table 3.2).

Removing the severe outliers (it should be noted that, for example, there are no lower bounds for those two instances), one gets an improved, although skewed distribution (skewness = 0.92) for the CPU time; median time is now 11 seconds and 75% CPU times are less than 36 seconds (as seen in Figure 3.10).

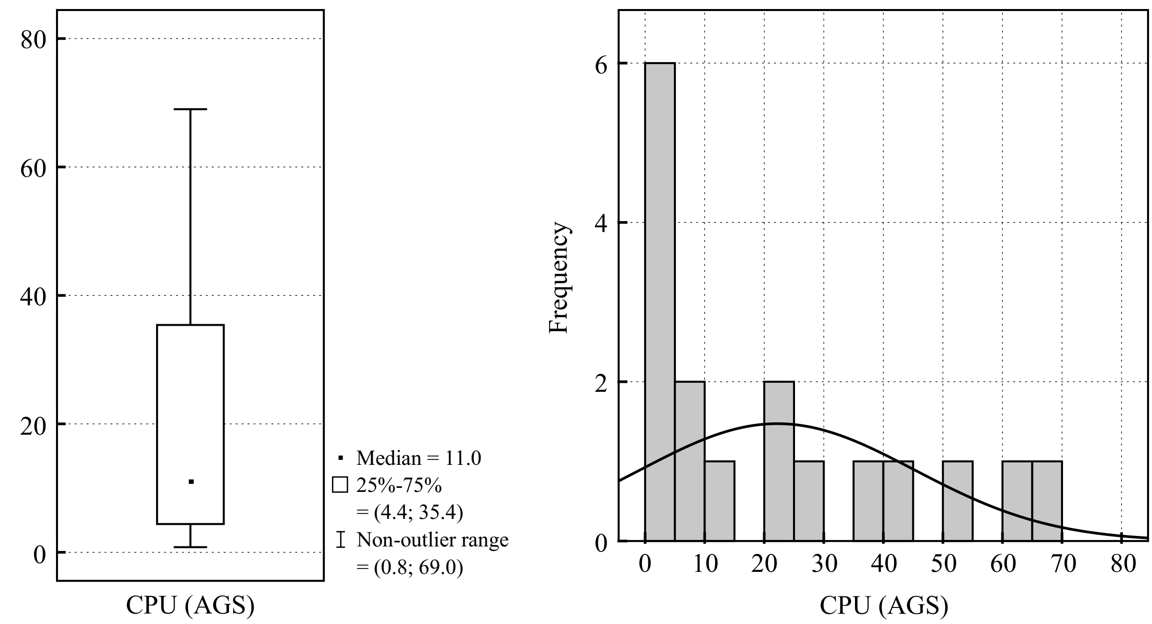


Figure 3.10 Boxplot (left) and histogram (right) for the CPU time, in seconds, without outliers (Table 3.2).

Figure 3.11 shows the linear relationship of the AGS metaheuristic CPU time and the number of clients (n), with regression bands for a level of confidence of 99% (regarding Table 3.2, without outliers).

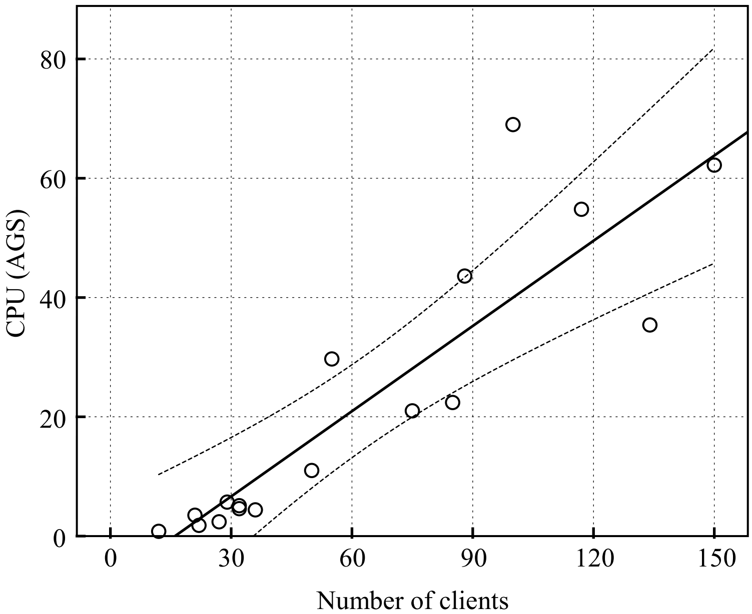


Figure 3.11 Relationship of the CPU time (in seconds) without outliers and the number of clients (Table 3.2).

Regarding the results for the instances by Barreto et al. (Table 3.2), 75% of the $\mathrm{Gap_{LB}}$ values are less than 3.5% (as seen in Figure 3.12). Moreover 8 over 17 values are less than 0.6%.

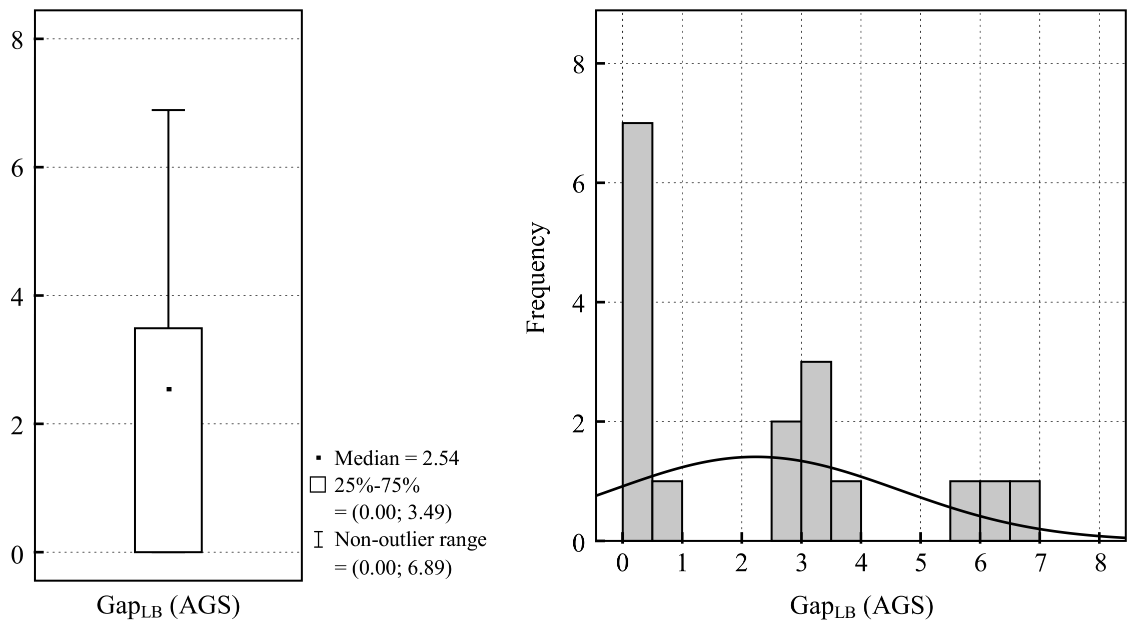


Figure 3.12 Boxplot (left) and histogram (right) for the $\mathrm{Gap_{LB}}$, in percentage, without outliers (Table 3.2).

The scatterplot, Figure 3.13, depicts two separate sets of values (A and B), pointing to the possibility of "low-level" quality lower bounds for the B set.

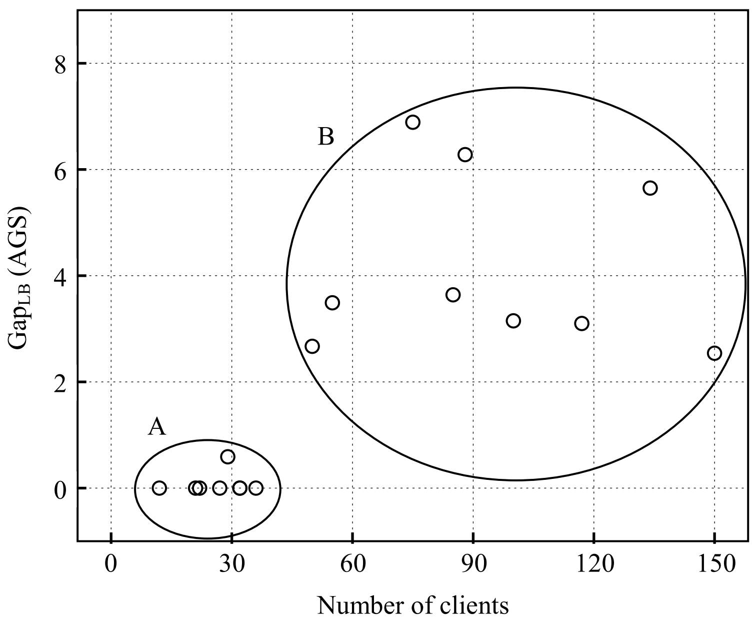


Figure 3.13 Scatterplot for the Gap_{LB} (in percentage) without outliers and the number of clients (Table 3.2).

Finally, it should be noted that for the 17 instances with known lower bound, the AGS metaheuristic was able to find 8 optimal values, to improve 5 ones, as well as to get 2 new upper bounds for those instances (Table 3.2).

Table 3.3 Results for the instances by Prins et al. (2006).

							AGS			
	Instance	n	m	Q	cl	BKR	Avg.	Best	CPU	Gap_{BKR}
1	20-5-0a	20	5	70	0	*54793	55517	54875	1.7	0.15
2	20-5-0b	20	5	150	0	*39104	40846	*39104	1.5	0.00
3	20-5-2a	20	5	70	2	*48908	49372	*48908	1.9	0.00
4	20-5-2b	20	5	150	2	*37542	*37542	*37542	1.3	0.00
5	50-5-0a	50	5	70	0	90111	92136	91975	5.4	2.07
6	50-5-0b	50	5	150	0	63242	67606	63307	9.4	0.10
7	50-5-2a	50	5	70	2	88298	90018	88457	13.0	0.18
8	50-5-2b	50	5	150	2	67340	68217	67496	13.6	0.23
9	50-5-3a	50	5	70	3	86203	87181	86203	18.1	0.00
10	50-5-3b	50	5	150	3	61830	62283	61830	9.2	0.00
11	100-5-0a	100	5	70	0	275993	280410	277539	39.3	0.56
12	100-5-0b	100	5	150	0	214392	216151	214818	34.2	0.20
13	100-5-2a	100	5	70	2	194598	197185	195342	20.9	0.38
14	100-5-2b	100	5	150	2	157173	158418	157200	24.4	0.02
15	100-5-3a	100	5	70	3	200246	203051	201340	57.1	0.55
16	100-5-3b	100	5	150	3	152586	155708	153265	23.1	0.44
17	100-10-0a	100	10	70	0	290429	318073	304274	22.5	4.77
18	100-10-0b	100	10	150	0	234641	271260	269812	20.5	14.99
19	100-10-2a	100	10	70	2	244265	247277	245671	53.1	0.58
20	100-10-2b	100	10	150	2	203988	206035	204391	102.7	0.20
21	100-10-3a	100	10	70	3	253344	257103	255223	35.0	0.74
22	100-10-3b	100	10	150	3	204597	206819	205177	45.6	0.28
23	200-10-0a	200	10	70	0	479425	483803	481123	186.2	0.35
24	200-10-0b	200	10	150	0	378773	381684	379192	195.0	0.11
25	200-10-2a	200	10	70	2	450468	454728	452760	157.6	0.51
26	200-10-2b	200	10	150	2	374435	377599	375710	72.2	0.34
27	200-10-3a	200	10	70	3	472898	480237	475055	155.9	0.46
28	200-10-3b	200	10	150	3	364178	367307	363946	113.4	-0.06
								Average	51.2	1.01
								Median	23.8	0.26

Figure 3.14 shows an asymmetric distribution (skewness = 1.42) as well as several moderate outliers (155.9; 157.6; 186.2; 195.0) for CPU time (Table 3.3). Median time is 23.75 seconds and 75% (Q3) of CPU times are less than 64.65 seconds.

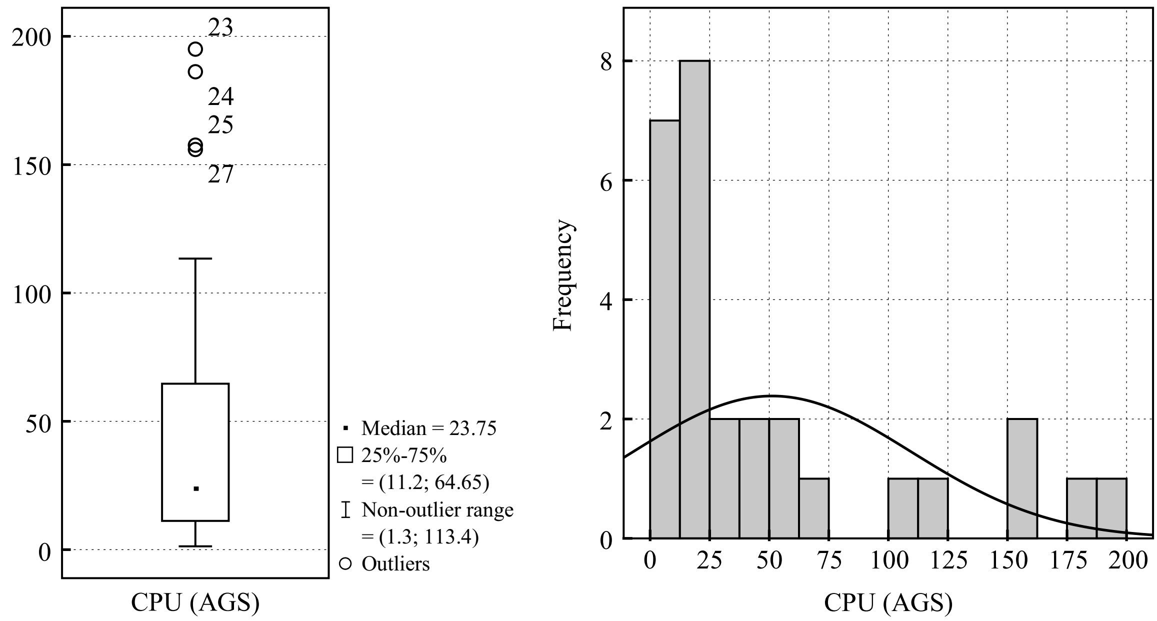


Figure 3.14 Boxplot (left) and histogram (right) for the CPU time in seconds (Table 3.3).

Figure 3.15 with severe outliers (2.07; 4.77; 14.99) suggests the use of the median value = 0.26% to characterize the location of $\mathrm{Gap_{BKR}}$ data (Table 3.3). Moreover, 75% of $\mathrm{Gap_{BKR}}$ values are less than 0.53%.

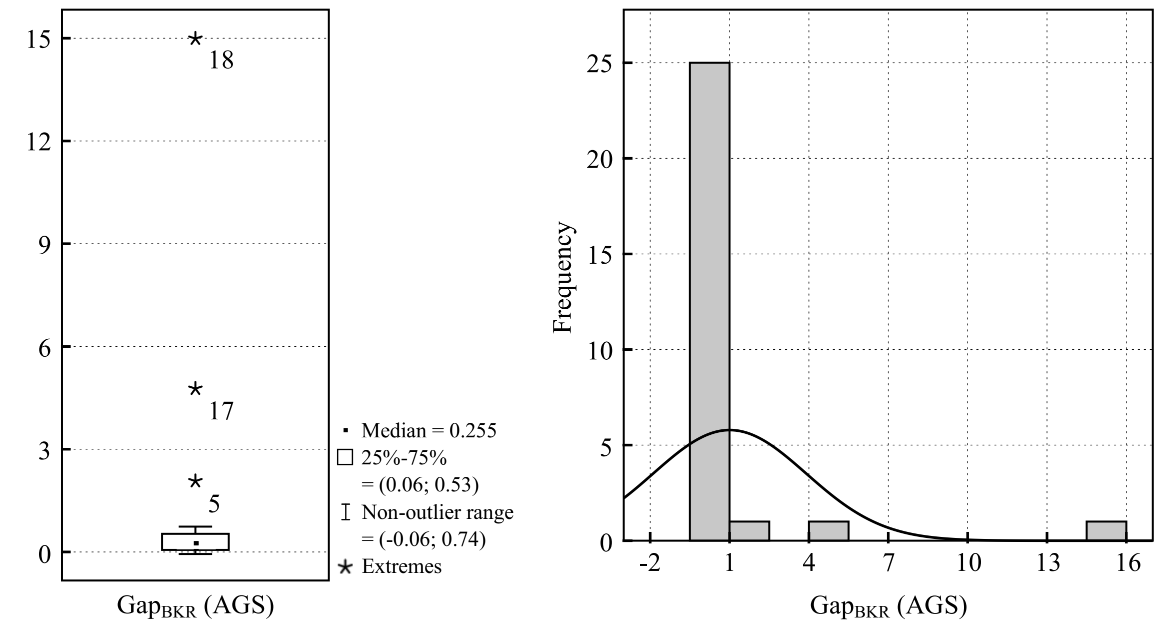


Figure 3.15 Boxplot (left) and histogram (right) for the $\mathrm{Gap_{BKR}}$ in percentage (Table 3.3).

Categorized boxplots in Figure 3.16 show the performance of the AGS metaheuristic regarding the number of clients (n) and depots (m), Table 3.3. $\mathrm{Gap_{BKR}}$ slightly increases with the number of clients (for $m = 5$); on the other hand, for $m = 10$ it decreases when the number of clients increases.

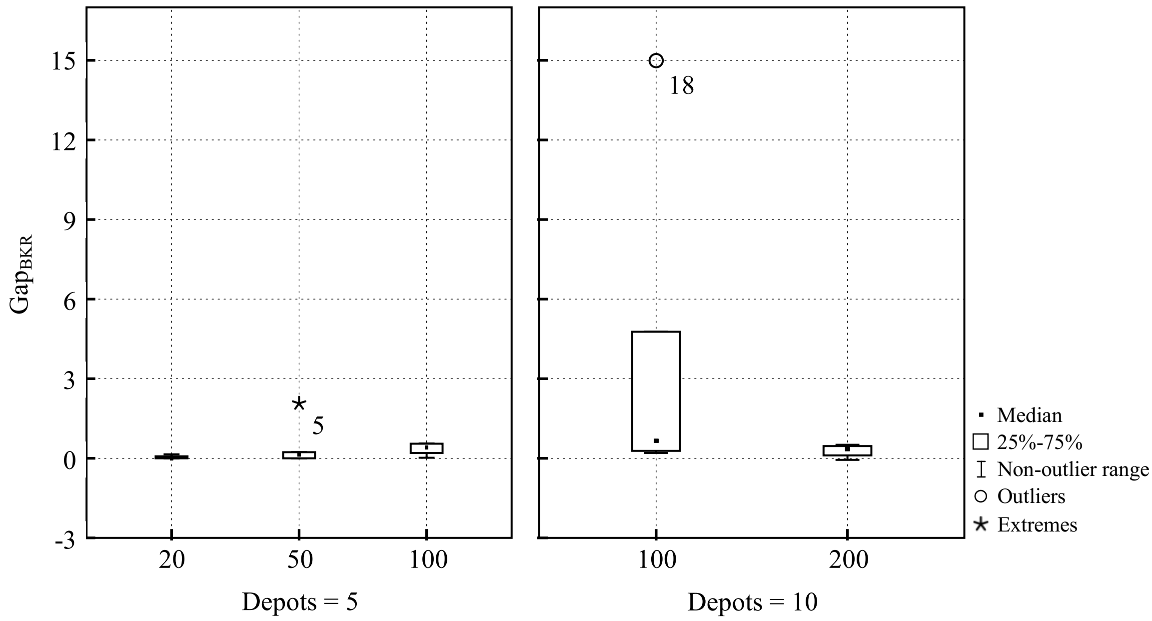


Figure 3.16 Categorized boxplots for the Gap$_{BKR}$ in percentage: clients = 20, 100, 100, and 200; depots = 5 and 10 (Table 3.3).

Using a dependent t-test for BKR and best AGS, the abovementioned good results, of the AGS metaheuristic, were statistically confirmed. The boxplots for both sets of values can be seen in Figure 3.17.

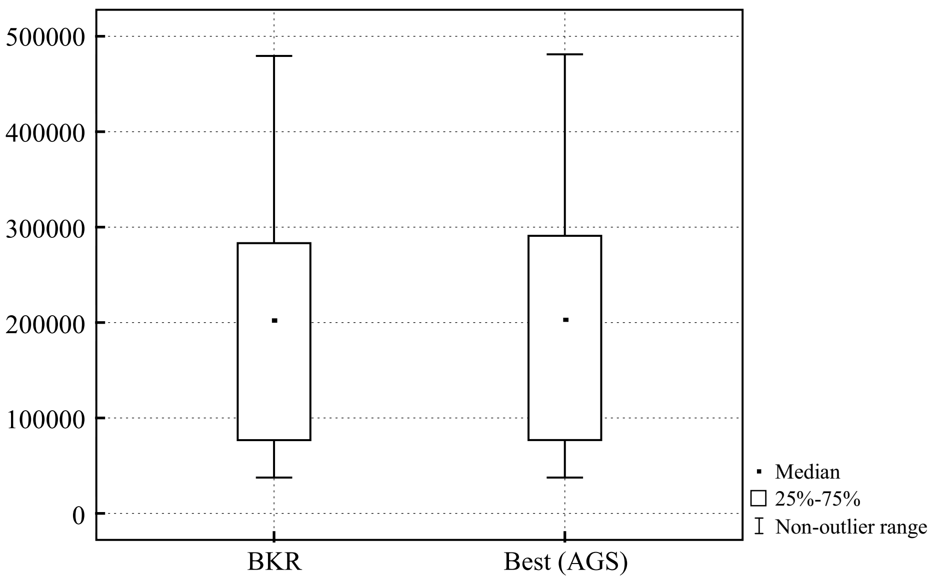


Figure 3.17 Boxplots for best known results (BKR) and best AGS results (Table 3.3).

The null hypothesis of equal means is not rejected for a significance level (α) of 5%, as the p-value $= 0.08 > \alpha = 0.05$.

On Table 3.4 a comparative analysis is performed, using the average results for the three sets of instances, between the best results of the proposed approach (AGS) and algorithms from the literature. CPU refers to the average computing time in seconds, Gap_{LB} to the average gap (in percentage) between the obtained results and the best known lower bound, and Gap_{BKR} represents the average gap to the currently best known results. The CPU time of the remaining algorithms was acquired from the corresponding publication.

Results suggest that the proposed metaheuristic obtains best average results (by around 1%) for the first two sets of instances and the second best result for the third set of instances. Regarding CPU time, LRGTS outperforms AGS, by presenting lower computing times while the GRASP has similar processing times (to be noted that both GRASP and LRGTS only run 13 out of the 19 instances from the second set, hence, considering the same 13 instances for AGS, the average CPU time is of 20.8 seconds and the Gap_{LB} of 2.14%).

Table 3.4 Average results for the three sets of instances.

	Tuzun and Burke		Barreto et al.		Prins et al.	
Algorithm	CPU	Gap_{BKR}	CPU	Gap_{LB}	CPU	Gap_{BKR}
TS (Tuzun and Burke, 1999)	11.5	3.74				
Cluster (Barreto et al., 2007)				4.94		
HybPSO (Marinakis and Marinaki, 2008b)			48.4	4.83		
Genetic (Marinakis and Marinaki, 2008a)				4.82		
GRASP (Prins et al., 2006)	159.6	2.92	†20.3	†3.23	103.2	3.58
LRGTS (Prins et al., 2007)	21.2	1.27	†17.6	†3.25	18.6	0.52
AGS	133.8	0.24	83.8	2.24	51.2	1.01

† Only 13 out of the 19 instances are considered.

Still, as the LRP is typically addressed at a strategic level and the average times to obtain the solutions range from one to two minutes, CPU time seems reasonable and serves only as a performance indicator among different methods.

3.2 Location-Arc Routing Problem

The vast majority of LRP papers address node routing. Nevertheless, one can consider several scenarios where the demand rather than being on the nodes of a network (usually a road network is assumed) is on the edges. These problems are referred to in the literature as location-arc routing problems (LARPs) and can derive from the capacitated arc routing problem (CARP).

It has been shown that the CVRP can be transformed into the CARP (Golden and Wong, 1981), and that the reverse is also possible, replacing each arc with three (Pearn et al., 1987) or two vertices (Baldacci and Maniezzo, 2006; Longo et al., 2006), making the two classes of problems equivalent. The same holds true for their location counterparts: the CLRP and the LARP.

For the three transformations of the CARP into the CVRP, the resulting instance requires either fixing of variables or the use of edges with infinite cost. Moreover, the resulting CVRP graph is a complete graph of larger size. Hence, with the transformation, the problem size increases and the planar structure of a usual CARP graph is lost (Wøhlk, 2008). The same can be extrapolated to the LARP.

This motivates the study of the LARP using dedicated methods and algorithms. As follows, a formal definition of this problem is given followed by some newly developed heuristic approaches and corresponding results, obtained on a new set of instances (Lopes et al., 2010b).

3.2.1 Problem Definition

The LARP, first introduced by Levy and Bodin (1989), consists of simultaneously determining depot location and routes in a graph in order to serve a specified set of required arcs under given operational constraints. Muyldermans (2003) has shown that, for this problem, an optimal solution exists with the facilities located on the vertices of the graph.

Formally, the LARP can be described on a weighted and directed graph $G = (V, A)$ with a vertex set V and a set of arcs A. The vertex set V contains a non-empty subset J of m potential depot locations ($J \subseteq V$) with a fixed cost f_j and capacity w_j associated ($j \in J$). Every arc $a = (i, j)$ in the arc set A has a traversal non-negative cost c_a and a non-negative demand for service d_a. The arcs with positive demand form the subset R of the arcs required to be serviced, only once, by a fleet K of identical vehicles with capacity Q. Vehicles start and end the route in the same depot, and each new vehicle (or route, as it is assumed that each vehicle performs a single route) involves a fixed cost F. The traversal of non-required arcs is known as "deadheading", being the associated cost denoted z_{ij} between any two vertices $i, j \in V$ (here, z_{ij} is the cost of the shortest path in G from i to j).

It is intended to determine the set of depots to be open in J and the tracing of the distribution routes assigned to each open depot in such a way that the sum of fixed and traversal costs to serve all arcs in R is minimized.

Assuming G to be connected, it is possible to transform it into a complete graph $\hat{G} = (\hat{V}, \hat{A})$ where $\hat{V}$ is composed of the set V_R of vertices containing the extremities of the arcs in R ($V_R \subseteq V$), and J ($\hat{V} = V_R \cup J$). As $\hat{G}$ is a complete graph and $V_R \subseteq \hat{V}$, R is a subset of $\hat{A}$. Each arc $a = (i, j)$ in the arc set $\hat{A}$ has a non-negative cost $\hat{c}_a$ which takes on the value c_a if $a \in R$, z_{ij} otherwise.

Let S be any subset of vertices in $\hat{V}$ ($S \subset \hat{V}$), $\delta^+(S)$ ($\delta^-(S)$) be the set of arcs leaving (entering) S, and $L(S)$ the set of arcs with both extremities in S. When S contains a single vertex v, $\delta^+(v)$ is a simplification for $\delta^+(\{v\})$. The following binary variables are used: x_{ak}, equal to one if arc $a \in \hat{A}$ is used in the route performed by vehicle $k \in K$; y_j, equal to one if depot j is to be opened; and y_{aj}, equal to one if the arc $a \in R$ is assigned to depot j. The LARP can be formulated as:

$$\text{(LARP)} \quad \min \quad Z = \sum_{j \in J} f_j y_j + \sum_{a \in A} \sum_{k \in K} \hat{c}_a x_{ak} + \sum_{k \in K} \sum_{a \in \delta^+(J)} F x_{ak} \tag{3.17}$$

$$\text{s.t.:} \quad \sum_{k \in K} x_{ak} = 1 \qquad \forall a \in R, \tag{3.18}$$

$$\sum_{a \in R} d_a x_{ak} \leq Q \qquad \forall k \in K, \tag{3.19}$$

$$\sum_{a \in \delta^+(i)} x_{ak} - \sum_{a \in \delta^-(i)} x_{ak} = 0 \qquad \forall i \in \hat{V}, \forall k \in K, \tag{3.20}$$

$$\sum_{a \in \delta^+(J)} x_{ak} \leq 1 \qquad \forall k \in K, \tag{3.21}$$

$$\sum_{a \in L(S)} x_{ak} \leq |S| - 1 \qquad \forall k \in K, \forall S \subseteq V_R, \tag{3.22}$$

$$\sum_{b \in \delta^+(j) \cap \delta^-(V_R)} x_{bk} + x_{ak} \leq 1 + y_{aj} \qquad \forall a \in R, \forall j \in J, \forall k \in K, \tag{3.23}$$

$$\sum_{a \in R} d_a y_{aj} \leq w_j y_j \qquad \forall j \in J, \tag{3.24}$$

$$x_{ak} \in \{0,1\} \qquad \forall a \in \hat{A}, \forall k \in K, \tag{3.25}$$

$$y_j \in \{0,1\} \qquad \forall j \in J, \tag{3.26}$$

$$y_{aj} \in \{0,1\} \qquad \forall a \in R, \forall j \in J. \tag{3.27}$$

The objective function (3.17) minimizes the sum of, respectively, the fixed costs of opening the depots, the costs of all traversed arcs, and the cost of acquiring vehicles. Constraints (3.18) ensure that each required arc is serviced once by exactly one vehicle. Capacity constraints are satisfied thanks to inequalities (3.19) and (3.24). Equalities (3.20) are the flow conservation constraints which, coupled with constraints (3.21), ensure the routes return to the departure depot. Constraints (3.22) are subtour elimination constraints while the set of constraints (3.23) specify that a required arc can be assigned to a depot only if there is a route linking them. Finally, constraints (3.25), (3.26), and (3.27) define the variables.

It can be noted that the LARP considered here can be seen as an extension of the CARP, where: multiple depots are considered and it is handled an additional level of decision (since the set of depots to install has to be obtained).

3.2.2 Recent Algorithmic Developments

As shown in Chapter 2, the LARP is one of the most overlooked in the literature. The great majority of papers in the LRP literature handles node routing, leaving the context of arc routing virtually untouched. As follows, a review on the current studies for the LARP will be presented.

The first work on the LARP, by Levy and Bodin (1989), intended to tackle a practical problem arising in the scheduling of postal carriers in the United States postal service. The developed algorithm used the location-allocation-routing (L-A-R) concepts described by Laporte (1988) for the LRP, which includes three steps: firstly, depots are to be located using a depot selection procedure; secondly, arcs with demand are to be allocated to depots; thirdly, an Euler tour route of minimum traverse cost is determined for each set of arcs allocated to depots.

Ghiani and Laporte (1999) addressed an undirected LARP, called location rural postman problem, in which depots are to be located and routes to be drawn (serving edges with demand), at minimum cost, in an undirected graph. The authors show the problem can be transformed into a rural postman problem if there is a single depot to open or no bounds on the number of depots. Using an exact branch-and-cut approach they solve the transformed problem.

On the work by Ghiani and Laporte (2001) a set of common applications for the LARP is mentioned (mail delivery, garbage collection and road maintenance). Furthermore the authors define the LARP as an extension of one of the three classical arc routing problems: the Chinese postman problem, the rural postman problem, and the CARP. The authors also present some insight on heuristic approaches using the decomposition of the problem into location (L), allocation (A) and routing (R) (Laporte, 1988): location-allocation-routing (L-A-R) and allocation-routing-location (A-R-L).

Muyldermans (2003) presents a variant of the LARP: the *p* dead-mileage problem. In this problem, unlike the previously addressed LARPs, splitting of the demand is allowed, that is, the client can be serviced more than once. Moreover, the objective is to minimize dead-mileage (deadheading).

Finally, the works by Pia and Filippi (2006) and Amaya et al. (2007) address variants of the CARP with a similar structure to the LARP, respectively, the CARP with mobile depots and the CARP with refill points. In the first, two different types of vehicles are considered: compactors and satellites. Compactors can be seen as moving depots for the satellites. The second problem considers vehicles servicing arcs that must be refilled at certain nodes (to be obtained) in order to continue its service.

From the previously mentioned variants, the LARP addressed here (and defined earlier) is the same studied by Ghiani and Laporte (1999) which can be seen as the arc routing equivalent to the CLRP, and thus an extension to the CARP.

3.2.3 Constructive Methods and Improvement Heuristics

The LARP, as mentioned earlier, results from the combination of a facility location problem and the CARP, both NP-hard problems and, as such, is NP-hard. Due to this, sharp bounds on the optimal value are typically hard to derive so, hardly big instances can be solved using exact methods. The best way to tackle these problems is then to use heuristic approaches, commonly used in practice (Hertz and Widmer, 2003), some of the most known being constructive methods and improvement heuristics.

Constructive methods are usually used to obtain initial solutions from which improvement heuristics can be used to attain better results. Furthermore, they are often used as the first step to many metaheuristic approaches. In this section constructive methods (extended augment-merge and extended merge) and improvement heuristics (reverse and relocate with both intra- and inter-route moves) are proposed to tackle the LARP.

Extended Augment-Merge

The augment-merge algorithm (illustrated in Figure 3.18) was proposed for the CARP by Golden and Wong (1981). It starts with a trivial solution in which each arc with demand is serviced by a separate route. Then, after sorting the obtained routes in decreasing cost order, it proceeds to the augment phase where, starting with the route with highest cost, it is seen if the route already goes through demand arcs on less costly routes. If so, and provided vehicle capacity is obeyed, the latter route is augmented into the former (the route with highest cost).

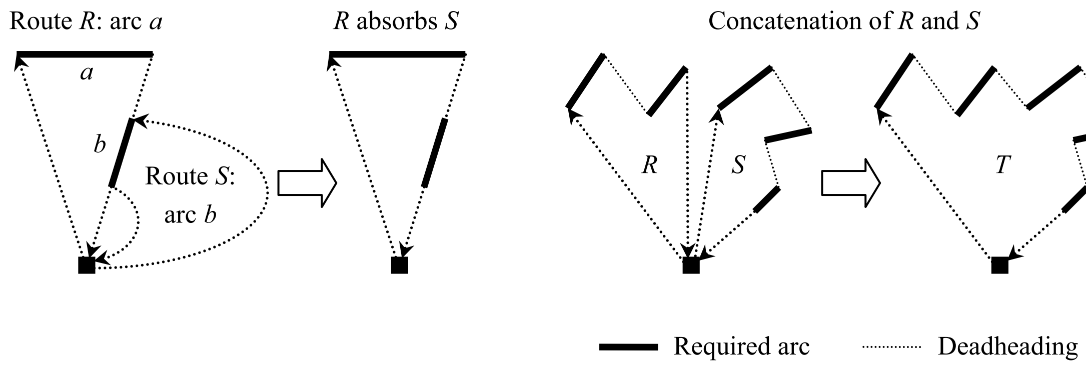


Figure 3.18 Augment (left) and merge (right) moves in augment-merge.

Afterwards the algorithm proceeds to the merge phase, where every feasible merge is evaluated for any two routes, merging the routes which provide the highest saving. This is done until no feasible saving exists. This last step is closely related to the well known "savings" or Clarke and Wright algorithm (Clarke and Wright, 1964).

The extended version (extended augment-merge – EAM) for the LARP obtains the initial solution by assigning each required arc to the closest depot in which they can fit, thus building a dedicated route. When all required arcs are assigned, the depots without demand to supply are closed. The augment phase is similar to augment-merge algorithm, increasingly considering depot capacity constraints. In the merge phase of the EAM, the resulting route T, which may result from four different merges, is evaluated for reassignment to all depots (totalling $4m$ possible merges). The resulting saving σ can be calculated as follows:

$$\sigma = F + z_{ri} + z_{jr} + z_{sk} + z_{ls} - z_{jk} - z_{ti} - z_{lt} + f_r\theta_r + f_s\theta_s - f_t(1 - y_t). \qquad (3.28)$$

θ_r (θ_s) is binary and equal to 1 if depot r (s), the depot of route R (S), supplies no more routes after the merger, and thus can be closed. y_t is a binary value (defined earlier for the formulation) equal to 1 if depot t (the depot currently evaluated to be assigned to T) is already open before the merge, and i, j, k, and l are the vertices of the arcs with demand which are connected to the depots in each route. The EAM ends when there is no more feasible merge with a positive saving.

Extended Merge

In the augment-merge algorithm it has been contested the use of the augment phase (Belenguer et al., 2006). If all the arcs in A are required to be serviced, it performs well, as the arcs absorbed by the higher cost routes are often contiguous. However, if it is not the case, the deadheading distance created between absorbed arcs cannot be recovered in the merge phase, leading to degraded results. Belenguer et al. (2006) further support this statement by presenting overall better results for the algorithm without the augment phase.

As the EAM derives from the augment-merge, a similar situation may occur. This suggests the development of an extended merge (EM) algorithm for the LARP, similar to the EAM, differing in not performing the augment phase, identical otherwise. Figure 3.19 depicts the merge phase used in both the EAM and the EM algorithms.

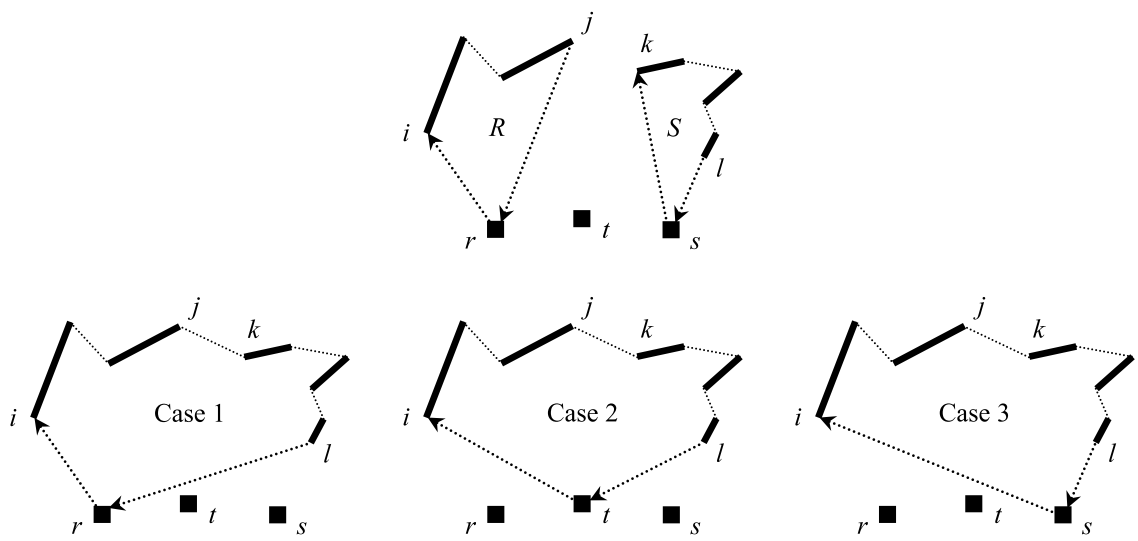


Figure 3.19 Some merges in the merge phase of the EAM and EM algorithms.

The extension of the merge phase (in both algorithms) uses some concepts from the extension to the savings algorithm proposed by Prins et al. (2006) for the CLRP (previously described in Section 3.1.3).

Reverse

The reverse improvement heuristic (Beullens et al., 2003) performs inside the routes and the corresponding move can be seen as the arc routing equivalent of the 2-opt move (Lin and Kernighan, 1973).

The reverse move is identical to the one used in the CARP (Figure 3.20). The algorithm performs by replacing a subsequence of arcs by the reverse, always insuring the required arcs are serviced. This may lead to other shortest paths (in the deadheading distance) linking the required arcs. The algorithm implements the first found feasible move that improves the solution. This is done sequentially until no more feasible moves can be found.

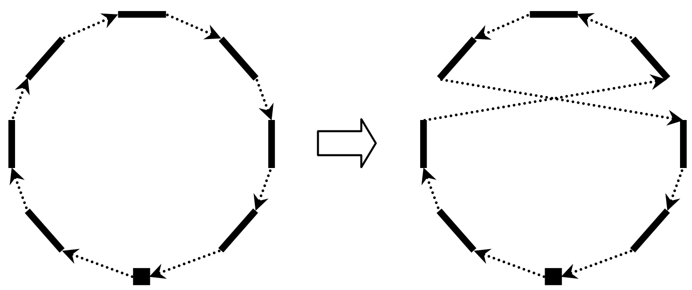


Figure 3.20 The reverse move applied to a route.

Relocate

In the relocate improvement heuristic (well known from the node routing context) two possible variations are considered: relocate inside the routes and relocate between two routes. In both variations the concept is to relocate a given arc (or subsequence of arcs) which requires service to another position in the route, or in another route.

The relocate algorithm inside the routes is based on the CVRP algorithm by Savelsbergh (1992), while the inter-route algorithm is based on the work by Beullens et al. (2003) for the CARP. In the latter, not only vehicle capacity but also depot capacity constraints have to be taken into account. In both algorithms the (subsequence of) required arc(s) or its reversal is reinserted, depending on which of two provides the biggest improvement.

3.2.4 Metaheuristics

Metaheuristics are general combinatorial optimization techniques that have rapidly demonstrated their usefulness and efficiency in solving hard problems (Glover and Kochenberger, 2003; Talbi, 2009). While in theory they can handle any combinatorial optimization problem, it is often the case that an important effort must be put on finding the right way to adapt the general parameters of these methods to the particular considered problem (Hertz and Widmer, 2003).

This is the case in this section, where three general metaheuristics (tabu search combined with a variable neighbourhood search – TS-VNS; greedy randomized adaptive search procedure – GRASP; and tabu search combined with a greedy randomized adaptive search procedure – TS-GRASP) are adapted and parameters tuned in order to tackle the LARP. The metaheuristics use the previously developed constructive and improvement methods in their specific framework.

Tabu Search-Variable Neighbourhood Search

This approach (TS-VNS) is an iterative framework composed of a tabu search (TS) and a variable neighbourhood search (VNS), respectively, for the location and (arc) routing phases. These two algorithms are performed iteratively until a stopping criterion is met, namely, a number $maxtsvns$ of iterations without improvements to the solution (empirically found to be equal to 10). The TS-VNS approach starts by obtaining a solution using the VNS without constraints on the subset of depots SD to use ($SD \subseteq J$).

Variable Neighbourhood Search. VNS is a metaheuristic proposed by Mladenović and Hansen (1997) in which the main concept is to perform a systematic change of neighbourhood within the local search. This is done by exploring increasingly distant neighbourhoods of the current solution. If an improvement is made, the search proceeds to the new solution and restarts the search. The steps of the basic VNS can be seen in Figure 3.21.

Initialization. Select the set of neighbourhood structures N_l, $l = 1,...,l_{max}$, that will be used in the search; find an initial solution S^i; choose a stopping condition.

Repeat the following steps until the stopping criteria is met.

(1) Set $l := 1$.

(2) Until $l = l_{max}$, repeat the following steps:

(a) *Shaking*. Generate a point S' at random from the l^{th} neighbourhood of S^i ($S' \in N_l(S^i)$).

(b) *Local search*. Apply some local search method with S^i as initial solution; denote with S'' the so obtained local optimum.

(c) *Move or not*. If this local optimum S'' is better than the incumbent S^i, move there ($S^i := S''$), and continue the search with N_1 ($l := 1$); otherwise, set $l := l + 1$.

Figure 3.21 Steps of the basic VNS (Hansen and Mladenović, 2001).

In the adopted VNS: N_l is a finite set of pre-selected neighbourhood structures inspired in the work by Polacek et al. (2008) for the CARP; the initial solution is obtained by performing the EAM (choice over the EM is due to being faster to obtain solutions, as the following steps behave similarly in both methods); and the stopping condition is a given number $maxvns$ of iterations reached (equal to the number of arcs in the problem, multiplied by ten: $maxvns = 10|A|$).

The shaking step uses the CROSS-exchange operator proposed by Taillard et al. (1997) for the VRP with time windows. It starts by randomly selecting two different routes, to which the CROSS-exchange operator is applied, swapping the sequence of consecutive required arcs. The number of required arcs which get swapped on each route is randomly obtained from an uniform distribution in the range [1, min(l, R_T)], R_T being the total number of required arcs for route T. When $l = l_{max}$ ($l_{max} = 6$, found empirically) the upper bound on the range is substituted by R_T. The described shaking step is biased to exchange smaller sequences of required arcs, while still allowing to perform large swaps.

The local search is applied solely on the two changed routes, and is composed of the formerly proposed reverse and relocate improvement heuristics, performing intra-route (reverse and relocate) and inter-route (relocate) improvements, sequentially, until no additional improvement can be found. The newly obtained local optimum is only accepted to move to if a cost reduction is obtained, thus making this a descent first improvement procedure which, according to Hansen and Mladenović (2001), can be easily transformed into a descent-ascent method (although experimental trials with this variation proved unfruitful for the present approach).

After obtaining the best solution S^* from the VNS, the TS-VNS approach proceeds to the location phase, performed by the TS algorithm.

Tabu Search. The used TS algorithm is the one presented by Filho and Galvão (1998), for the concentrator location problem, which provides near-optimal results in reduced CPU time, validating its use. TS was proposed by Glover (1986) and has become one of the most widespread local search methods for combinatorial optimization. It uses a working memory called tabu list in which some attributes are stored (and forbidden to be used) for a number of moves.

In the used TS algorithm some adaptations were made to handle the LARP. Each route in the current best solution is collapsed into a single client (regarding demand), and the distance to the several feasible depots is the smallest insertion cost of the depot in the route, as in Barreto et al. (2007). The problem thus becomes a facility location problem and the algorithm tries to obtain the best depot location for the current routes. Using the best obtained depot configuration (excluding the remaining depots from the problem), the TS-VNS approach proceeds to constructing the routes (routing phase) using the abovementioned VNS algorithm. If no improvement was found in the last (five) iteration(s), the approach provides some diversification by randomly opening one (two) depot(s) and closing another from SD, always insuring depot capacity constraints are obeyed.

A flowchart depicting the TS-VNS approach can be seen in Figure 3.22.

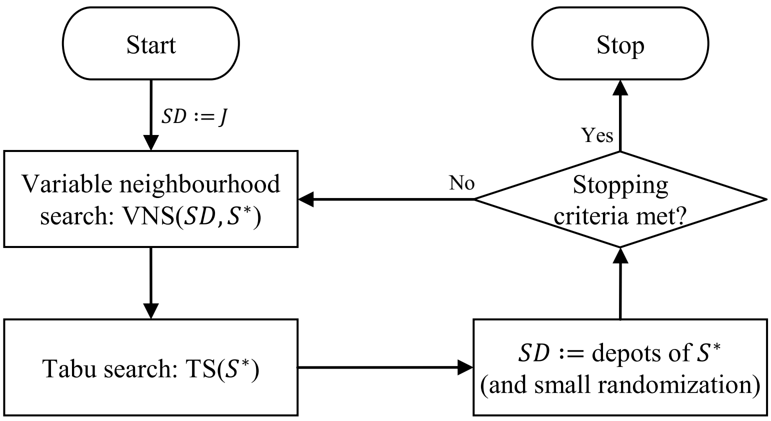


Figure 3.22 Flowchart of the TS-VNS metaheuristic for the LARP.

Greedy Randomized Adaptive Search Procedure

In order to obtain the GRASP, the EAM and the EM are randomized to provide the required greedy heuristic. Both methods are thus used in the constructive algorithm of the metaheuristic, giving place to two variations, based respectively on the two methods: GRASP-EAM and GRASP-EM. The only difference between the two variations is the constructive algorithm (which can or not have the augment phase). The following features are valid for both variations and apply some concepts of the GRASP developed by Prins et al. (2006) for the CLRP.

For the randomized version of the constructive algorithms a restricted candidate list (RCL) of size φ was created from the savings calculated during the merge evaluations. The RCL contains the pairs of routes providing the best φ savings, from which one is randomly chosen to be performed. Changing the size of the RCL during the GRASP has been shown to often give better results (Resende and Ribeiro, 2003). So, at each merge, φ is randomly selected in [1, φ_{max}], where φ_{max} is the maximum RCL size allowed (found empirically: $\varphi_{max} = 7$).

The local search (LS), typically used in GRASP metaheuristics, is based on the reverse and relocate improvement heuristics previously presented, which performs intra-route (reverse and relocate) and inter-route (relocate) improvements, sequentially, until no additional improvement can be found.

Moreover, a learning process was included in the GRASP which reduces the computational time and improves the final solution (Prins et al., 2006). The constructive algorithms, at each iteration it, provide a solution S^{it} which often has many open depots and, although the merges can close some, it may not be enough. In order to obtain better solutions, a subset SD of available depots is chosen to be used in the constructive algorithms ($SD \subseteq J$). In the first iteration of the GRASP all depots are used, then, one of them is iteratively picked from J. The remaining depots in SD are randomly chosen (both the number to open and which) at each iteration, always insuring there is enough capacity to service all clients. This can be seen as a diversification.

Adding a memorization during the diversification mode enables the GRASP to learn about the good subset to open, and to possibly find better solutions. An intensification mode using this learning process was implemented, varying the GRASP iterations (using the boolean value $divmode$ and reaching the maximum of $maxit = 75$) between:

- Diversification mode – applied for $maxdiv = 8$ iterations, in which the solution space is explored by varying the subset of open depots (explained previously).
- Intensification mode – performed for $maxint = 7$ iterations, where an attempt is made to improve the routing for the selected depots (SD), obtained from the currently best found solution.

The parameters were set after a preliminary experimenting phase and allow five complete runs of the two modes returning, in the end, the best found solution (S^*). Figure 3.23 is a flowchart of the presented GRASP.

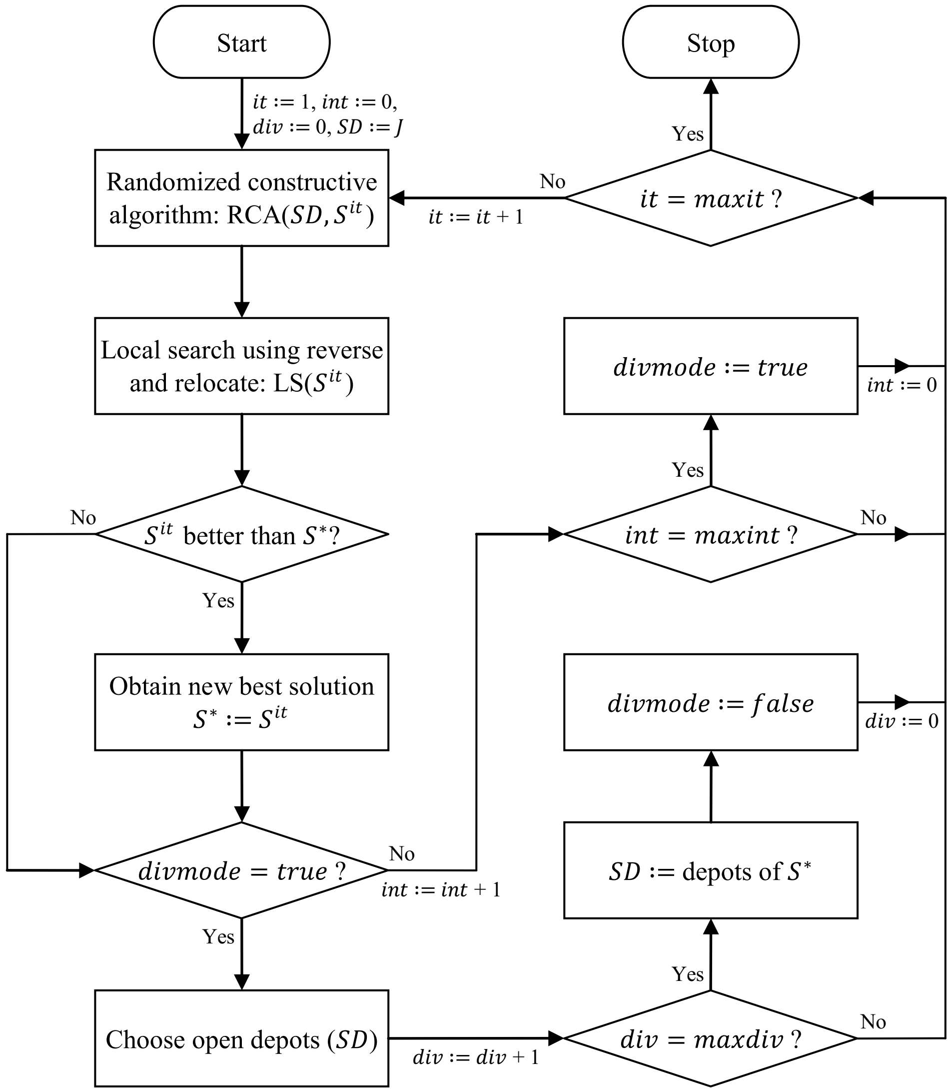


Figure 3.23 Flowchart of the GRASP metaheuristic for the LARP.

As stated previously, this GRASP leads to two variations (which will be evaluated in the comparative analysis): GRASP-EAM, using the randomized version of the EAM method; and GRASP-EM, randomizing the constructive method EM.

Tabu Search-Greedy Randomized Adaptive Search Procedure

The approach presented in this section, the TS-GRASP, combines some aspects of the previously described TS and GRASP. The TS handles the location phase while the GRASP addresses the (arc) routing phase. This is an attempt to integrate the best features of the previous approaches. An iterative framework is used, in which the algorithms for both phases are performed iteratively until a stopping criterion is met (a number $maxtsgrasp$ of iterations without improvements to the solution, empirically found: $maxtsgrasp = 10$). The required adaptations will be described as follows.

Similarly to the TS-VNS approach, the TS-GRASP starts by obtaining a solution for the routing phase without constraints on the subset of depots SD to use ($SD \subseteq J$), however, instead of the VNS, the GRASP is used.

Greedy Randomized Adaptive Search Procedure. This GRASP uses the EM for the randomized constructive algorithm, as it provides bigger diversity and better results (see Section 3.2.5), returning a solution S^{it} at each iteration it. The randomization performs similarly, using the RCL (with the same value of φ_{max}), as does the LS, with the reverse and relocate improvement heuristics (applied sequentially while there is improvement to the solution). Unlike the previous GRASP however, there is no diversification mode ($maxdiv = 0$), as the TS handles the location phase. Hence, this GRASP tries to obtain the best results (always in intensification mode) for the same given depot configuration (SD).

The $maxit$ parameter also differs from the previous approach, varying at each iteration according to:

$$maxit = \lceil 0.25 ittsgrasp \rceil + 1 \tag{3.29}$$

where $ittsgrasp$ is the number of iterations without improvement. This allows to intensify the search as less improvements are found (insuring the GRASP is performed at least once).

The best obtained solution (S^*) is used in the location phase (TS algorithm) to where the TS-GRASP approach proceeds once the maximum number of iterations $maxit$ is reached.

Tabu Search. The TS algorithm in this approach works exactly as in the TS-VNS. Routes are collapsed into a single client and distances to depots are updated. Using the algorithm by Filho and Galvão (1998), the best subset SD of depots to open is obtained, to which is added a small diversification: obeying to capacity constraints, if no improvement was found in the last (five) iteration(s), randomly, one (two) depot(s) is (are) chosen to be open and one to be closed.

The TS-GRASP approach then continues again to the routing phase, restarting the GRASP and using only the new subset of depots SD, as seen in Figure 3.24.

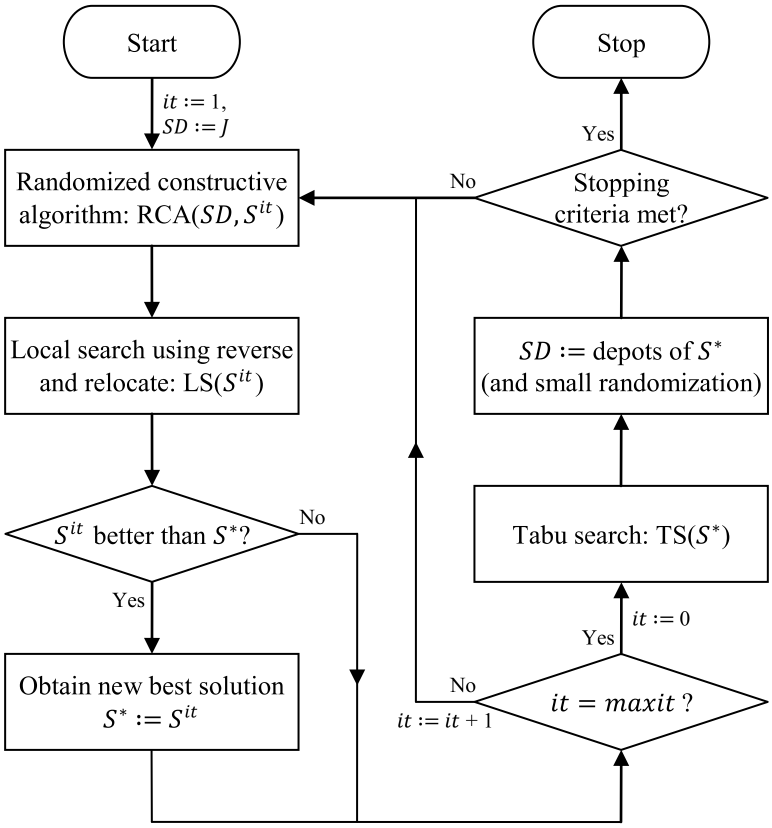


Figure 3.24 Flowchart of the TS-GRASP metaheuristic for the LARP.

3.2.5 Computational Results

In order to ascertain the performance of the methods and approaches proposed for the LARP, experimental results are obtained. Firstly, implementation issues are described and a new set of benchmark instances is proposed. Secondly, a comparative analysis is performed on the algorithmic proposals for the LARP using the newly devised instances.

Implementation and Benchmark Instances

All the aforementioned approaches were implemented in C# and the results obtained using a 3.00 GHz Intel Xeon E5450 Quad Core CPU with 8 GB of RAM and Windows XP (without parallel processing). For obtaining the deadheading distance (z_{ij}), in both the constructive methods and improvement heuristics (and consequently on the metaheuristic approaches), the well known Dijkstra's algorithm (Dijkstra, 1959) was used.

In order to compare results and times a new set of instances is proposed (increasingly justified by the absence of benchmark instances in the literature). The instances were drawn from the CARP literature (the original sets can be found in http://www.uv.es/belengue/carp.html) and adapted to the LARP. These are the well known and widely used instances from: Golden et al. (1983), named "gdb"; Benavent et al. (1992), called "bccm"; and Eglese (1994), named "eglese". The first two sets were generated on a graph, with all the edges being required edges, while the latter originates from a real-world (winter gritting) problem for the road network of Lancashire (UK), where two graphs were obtained and the different instances created by changing the set of required edges and the capacities of the vehicles.

From the referred instances for the CARP, some instances were chosen (promoting diversity) and adapted to support more than one depot and a cost structure in which location costs range 30-70% of the total cost (with distribution costs regarding deadheading distance). Table 3.5 displays the data regarding the proposed instances named after the original CARP set followed by the instance number/name in the original set (e.g. eglese-E1-A is the instance "E1-A" from Eglese, 1994). The first columns of the table display the name of the instance and the cardinality of the vertex (V), edges (E), and required edges (R) sets (all instances work with undirected graphs). Then follows the data regarding the depots, namely, the number of potential depot locations (m) and the average depot fixed cost ($\bar{f}$). Finally, columns "Q" and "F" refer, respectively, to the vehicles capacity and fixed cost. The proposed instances are available at http://lore.web.ua.pt/.

Table 3.5 Data regarding the proposed instances for the LARP.

	Instance	$\|V\|$	$\|E\|$	$\|R\|$	m	$\bar{f}$	Q	F
1	gdb-20	11	22	22	3	10	27	1
2	gdb-22	11	44	44	3	5	27	1
3	gdb-1	12	22	22	3	20	5	5
4	gdb-2	12	26	26	3	25	5	5
5	bccm-1B	24	39	39	5	15	120	2
6	bccm-2C	24	34	34	5	50	40	2
7	bccm-8A	30	63	63	5	10	200	5
8	bccm-6B	31	50	50	5	40	120	5
9	bccm-7A	40	66	66	5	5	200	2
10	bccm-4C	41	69	69	5	25	130	2
11	bccm-9B	50	92	92	5	30	175	5
12	bccm-10D	50	97	97	5	20	75	5
13	eglese-E1-A	77	98	51	10	250	305	20
14	eglese-E2-B	77	98	72	10	750	200	20
15	eglese-E3-C	77	98	87	10	1000	135	50
16	eglese-S1-A	140	190	75	10	400	210	50
17	eglese-S2-B	140	190	147	10	2500	160	100
18	eglese-S3-C	140	190	159	10	1500	120	100

Comparative Analysis

Results for the constructive methods (with and without the local search – LS – composed by the improvement heuristics applied sequentially until no more improvements can be found) and for the metaheuristic approaches were obtained for the newly devised instances. For the constructive methods, a single run for each instance is performed (as there is no randomization), and the corresponding result and computing time found. For the metaheuristic approaches, twenty runs were made for each instance, from which was acquired the average and best result (and the time to obtain it).

Results for the constructive methods (with and without LS) can be seen in Tables 3.6 and 3.7, respectively, for the EAM and EM. Regarding the metaheuristic approaches (TS-VNS, GRASP-EAM, GRASP-EM, and TS-GRASP) results are shown in Tables 3.8 and 3.9. In the tables, the first columns display the instances name, followed by the overall best result (BR). Afterwards, it is shown, for each algorithm: the obtained results – Result (or average results – Avg. – and best results – Best – for the metaheuristic approaches); CPU time, in seconds; and the gap (Gap_{BR}), in percentage, between the overall best result and the algorithm (best) result. Similarly to the CLRP, for each of the tables (in the LARP) average and median values are provided as data showed, for CPU time and Gap_{BR}, heavily skewed distributions and/or outlying data points.

Table 3.6 Results for the EAM constructive method with and without LS.

	Instance	BR	EAM			EAM + LS		
			Result	CPU	Gap_{BR}	Result	CPU	Gap_{BR}
1	gdb-20	135	154	0.00	14.07	148	0.00	9.63
2	gdb-22	214	229	0.00	7.01	229	0.00	7.01
3	gdb-1	353	403	0.00	14.16	390	0.00	10.48
4	gdb-2	390	462	0.00	18.46	447	0.00	14.62
5	bccm-1B	211	296	0.00	40.28	294	0.00	39.34
6	bccm-2C	370	487	0.00	31.62	487	0.00	31.62
7	bccm-8A	424	550	0.01	29.72	519	0.01	22.41
8	bccm-6B	329	488	0.00	48.33	488	0.00	48.33
9	bccm-7A	297	355	0.01	19.53	338	0.01	13.80
10	bccm-4C	458	563	0.01	22.93	543	0.01	18.56
11	bccm-9B	406	601	0.01	48.03	570	0.02	40.39
12	bccm-10D	546	630	0.02	15.38	630	0.02	15.38
13	eglese-E1-A	2985	5157	0.01	72.76	4679	0.01	56.75
14	eglese-E2-B	5480	11237	0.02	105.05	11081	0.02	102.21
15	eglese-E3-C	8643	14630	0.02	69.27	14606	0.02	68.99
16	eglese-S1-A	4315	7369	0.02	70.78	6806	0.02	57.73
17	eglese-S2-B	15069	33970	0.08	125.43	33919	0.08	125.09
18	eglese-S3-C	16029	26020	0.08	62.33	25869	0.09	61.39
			Average	0.02	45.29		0.02	41.32
			Median	0.01	35.95		0.01	35.48

Looking at the results for the constructive methods, it can be concluded that, overall, EM performs better than EAM. This may lead to conclude that, the claim by Belenger et al. (2006) for the CARP, suggesting the augment phase generally leads to poorer results, is valid for the LARP. However, as the EAM obtains faster results (and with somewhat similar final results) it may be an interesting constructive method to be used in metaheuristics.

Table 3.7 Results for the EM constructive method with and without LS.

	Instance	BR	EM Result	EM CPU	Gap_{BR}	EM + LS Result	EM + LS CPU	Gap_{BR}
1	gdb-20	135	160	0.00	18.52	160	0.00	18.52
2	gdb-22	214	223	0.01	4.21	223	0.01	4.21
3	gdb-1	353	389	0.00	10.20	389	0.00	10.20
4	gdb-2	390	448	0.00	14.87	448	0.00	14.87
5	bccm-1B	211	277	0.01	31.28	273	0.01	29.38
6	bccm-2C	370	533	0.00	44.05	529	0.00	42.97
7	bccm-8A	424	550	0.03	29.72	534	0.03	25.94
8	bccm-6B	329	496	0.01	50.76	494	0.01	50.15
9	bccm-7A	297	366	0.03	23.23	353	0.03	18.86
10	bccm-4C	458	564	0.04	23.14	550	0.04	20.09
11	bccm-9B	406	574	0.09	41.38	562	0.09	38.42
12	bccm-10D	546	672	0.10	23.08	661	0.10	21.06
13	eglese-E1-A	2985	4520	0.03	51.42	4424	0.03	48.21
14	eglese-E2-B	5480	9166	0.07	67.26	9058	0.07	65.29
15	eglese-E3-C	8643	14696	0.11	70.03	14542	0.11	68.25
16	eglese-S1-A	4315	5152	0.11	19.40	4847	0.11	12.33
17	eglese-S2-B	15069	31626	0.61	109.87	31516	0.62	109.14
18	eglese-S3-C	16029	24626	0.72	53.63	24552	0.72	53.17
			Average	0.11	38.11		0.11	36.17
			Median	0.03	30.50		0.03	27.66

Table 3.8 Results for the TS-VNS and the GRASP (using EAM) metaheuristic approaches.

	Instance	BR	TS-VNS Avg.	TS-VNS Best	TS-VNS CPU	Gap_{BR}	GRASP-EAM Avg.	GRASP-EAM Best	GRASP-EAM CPU	Gap_{BR}
1	gdb-20	135	139	139	0.36	2.96	136	135	0.03	0.00
2	gdb-22	214	218	216	1.38	0.93	217	215	0.29	0.47
3	gdb-1	353	366	359	0.61	1.70	363	353	0.02	0.00
4	gdb-2	390	400	400	0.47	2.56	405	400	0.04	2.56
5	bccm-1B	211	222	220	1.19	4.27	221	216	0.09	2.37
6	bccm-2C	370	384	384	0.77	3.78	385	384	0.04	3.78
7	bccm-8A	424	445	433	5.41	2.12	436	427	0.56	0.71
8	bccm-6B	329	335	335	1.62	1.82	339	336	0.14	2.13
9	bccm-7A	297	312	304	4.69	2.36	309	302	0.44	1.68
10	bccm-4C	458	473	467	5.51	1.97	471	458	0.52	0.00
11	bccm-9B	406	417	415	5.60	2.22	419	413	1.30	1.72
12	bccm-10D	546	568	553	6.51	1.28	557	549	1.63	0.55
13	eglese-E1-A	2985	3275	3256	6.16	9.08	3332	3231	0.43	8.24
14	eglese-E2-B	5480	5888	5811	4.97	6.04	6095	5856	0.87	6.86
15	eglese-E3-C	8643	9194	9147	5.07	5.83	9290	9123	1.42	5.55
16	eglese-S1-A	4315	4692	4586	8.06	6.28	4699	4404	1.10	2.06
17	eglese-S2-B	15069	15989	15820	33.21	4.98	16109	15592	6.76	3.47
18	eglese-S3-C	16029	17273	17090	40.96	6.62	17471	16933	7.88	5.64
				Average	7.36	3.71			1.31	2.66
				Median	5.02	2.76			0.48	2.10

Regarding the metaheuristic approaches, computing times are, on average, less than 10 seconds, being the GRASP-EAM the fastest, followed by the TS-GRASP, the TS-VNS, and the GRASP-EM (albeit the difference between them is negligible as this is a strategic problem).

Table 3.9 Results for the GRASP (using EM) and the TS-GRASP metaheuristic approaches.

	Instance	BR	GRASP-EM				TS-GRASP			
			Avg.	Best	CPU	Gap_{BR}	Avg.	Best	CPU	Gap_{BR}
1	gdb-20	135	136	135	0.10	0.00	137	135	0.04	0.00
2	gdb-22	214	217	214	0.74	0.00	217	214	0.29	0.00
3	gdb-1	353	359	359	0.10	1.70	360	353	0.05	0.00
4	gdb-2	390	394	390	0.17	0.00	397	390	0.11	0.00
5	bccm-1B	211	216	211	0.59	0.00	218	211	0.39	0.00
6	bccm-2C	370	382	370	0.35	0.00	374	370	0.17	0.00
7	bccm-8A	424	434	425	3.07	0.24	438	424	0.97	0.00
8	bccm-6B	329	336	329	1.38	0.00	339	329	0.64	0.00
9	bccm-7A	297	303	297	3.01	0.00	309	301	1.18	1.35
10	bccm-4C	458	466	458	3.35	0.00	465	458	1.36	0.00
11	bccm-9B	406	412	406	9.52	0.00	414	408	4.78	0.49
12	bccm-10D	546	560	552	11.17	1.10	559	546	3.83	0.00
13	eglese-E1-A	2985	3166	3014	2.07	0.97	3139	2985	2.14	0.00
14	eglese-E2-B	5480	5730	5584	5.29	1.90	5649	5480	2.20	0.00
15	eglese-E3-C	8643	8738	8643	10.31	0.00	8819	8665	4.13	0.25
16	eglese-S1-A	4315	4535	4378	8.16	1.46	4438	4315	2.78	0.00
17	eglese-S2-B	15069	15602	15307	55.04	1.58	15400	15069	22.64	0.00
18	eglese-S3-C	16029	16587	16109	65.01	0.50	16251	16029	24.21	0.00
				Average	9.97	0.52			4.00	0.12
				Median	3.04	0.00			1.27	0.00

Concerning the gap to best results (Gap_{BR}), both the line plots and boxplots in Figure 3.25 suggest two overall groups (A and B) with different performance: constructive methods (EAM and EM with and without LS) and metaheuristic approaches (TS-VNS, GRASP-EAM, GRASP-EM, and TS-GRASP).

Using a correspondence analysis (CA) to analyse overall results of the algorithms, the two-dimensional scatterplot for axis 1 and 2 (accounting for 99% of inertia), Figure 3.26, clearly distinguishes three groups (I, II, III), further splitting constructive methods. It should be noted that dimension 1 (with 86.44% of inertia), being the most important dimension, separates and confirms the suggested two overall groups (A and B).

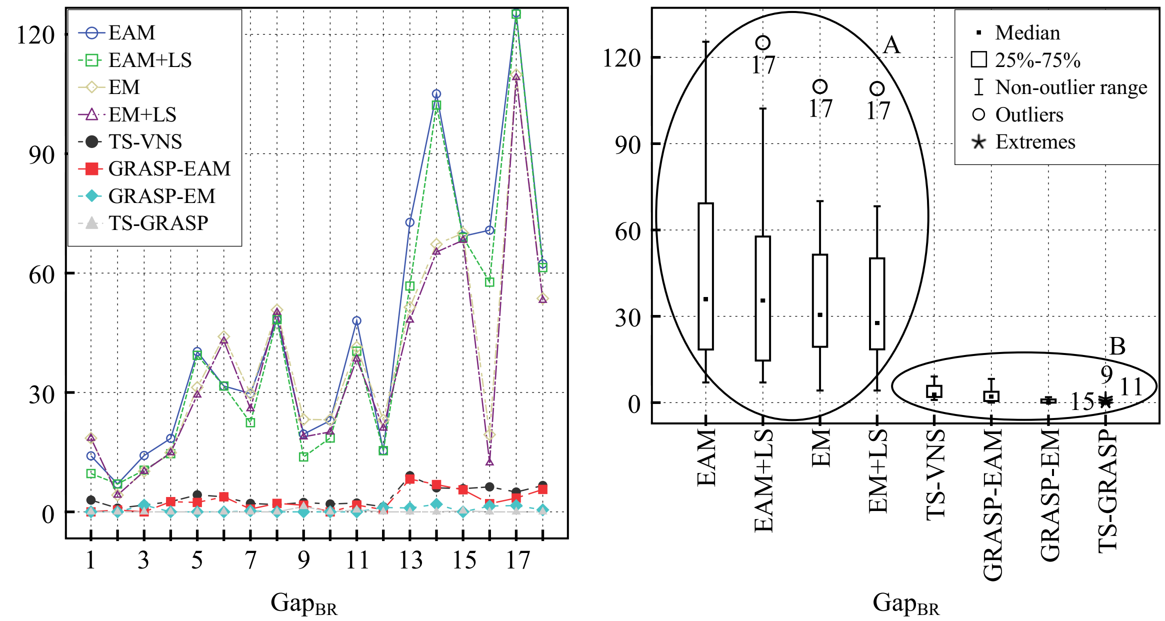


Figure 3.25 Line plots (left) and boxplots (right) for the Gap$_{BR}$, in percentage, concerning constructive methods (EAM and EM with and without LS) and metaheuristic approaches (TS-VNS, GRASP-EAM, GRASP-EM, and TS-GRASP), Tables 3.6, 3.7, 3.8, and 3.9.

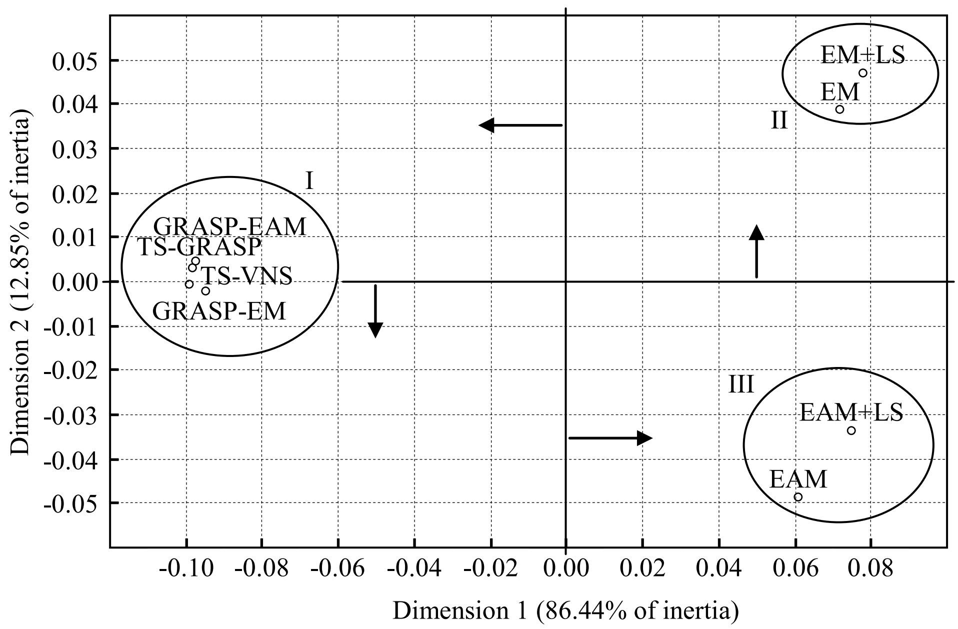


Figure 3.26 Two-dimensional scatterplot for axis 1 and 2 (accounting for 99% of inertia), for results concerning: EAM, EAM+LS, EM, EM+LS, TS-VNS, GRASP-EAM, GRASP-EM, and TS-GRASP (Tables 3.6, 3.7, 3.8, and 3.9).

Exploratory data analysis (EDA) suggests that metaheuristic approaches perform better, to this set of instances, than constructive methods. Thus, in Figure 3.27, the use of correspondence analysis and cluster analysis for results concerning just metaheuristics and best results, indicates that TS-GRASP is the most similar to best results (BR).

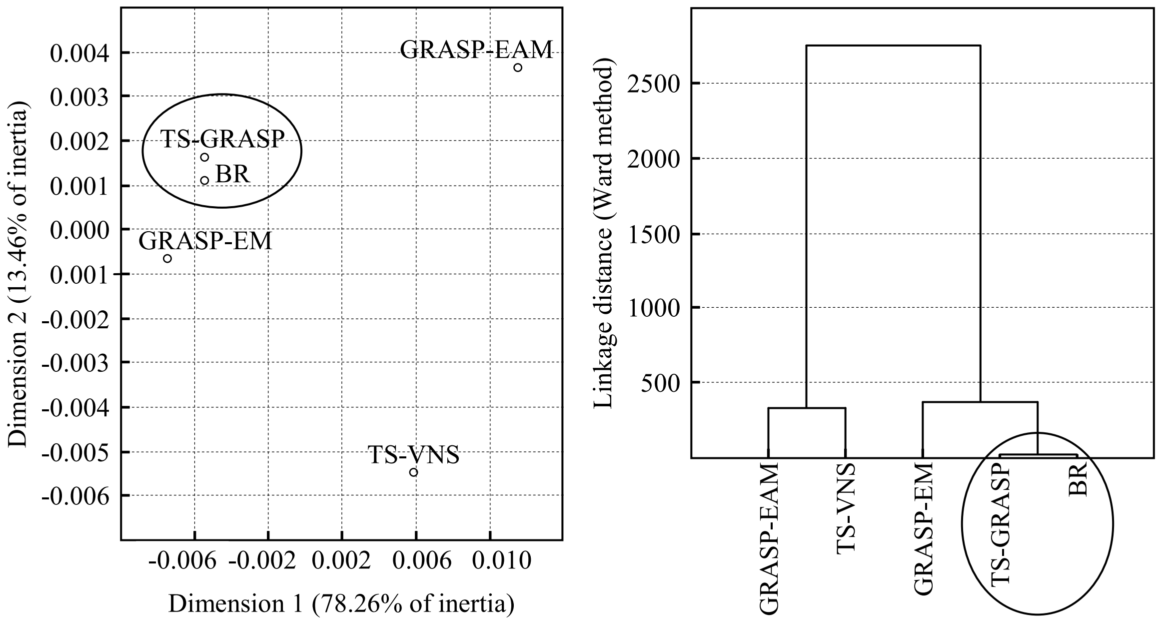


Figure 3.27 Two-dimensional correspondence analysis scatterplot (left) and cluster analysis dendogram (right) for results concerning: BR, TS-VNS, GRASP-EAM, GRASP-EM, and TS-GRASP (Tables 3.8 and 3.9).

The lack of normality (depicted in the boxplots of Figure 3.28) as well as the small data size (18 instances) strongly advise the use of a non-parametric test. Therefore, the abovementioned good results for the TS-GRASP metaheuristic, were statistically confirmed using the Wilcoxon matched pairs test for BR and best TS-GRASP results. The null hypothesis of equal median is not rejected for a significance value (α) of 5%, as the p-value = 0.11.

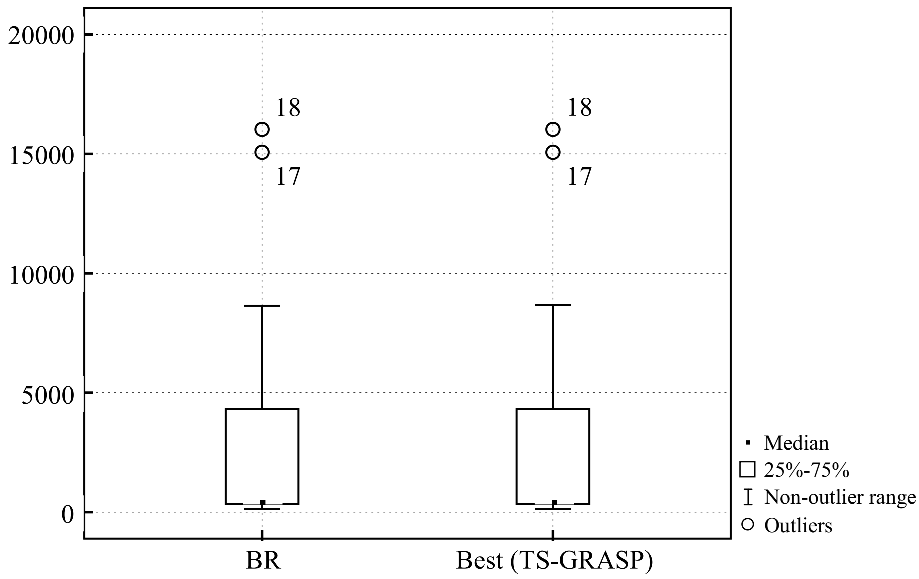


Figure 3.28 Boxplots for best results (BR) and best TS-GRASP results (Table 3.9).

The TS-VNS approach is the one which presents the worst results which, looking at the results of the TS-GRASP, leads to conclude that this is due to the VNS (used in the routing phase) not performing as well as the GRASP on the route building/improvement. Comparing the results of the two GRASP (GRASP-EAM and GRASP-EM), GRASP-EM performs better by providing an average as well as a median improvement of around 2%. This may be due to, by not having an augment phase, the approach allows for bigger diversity in the possible merges (inside the randomized constructive algorithm), thus facilitating the finding of better results.

Looking at the TS-GRASP, when compared to the GRASP-EM results, it can be concluded that the TS, used in the location phase, performs better than the diversity mechanism used in the choosing of open depots in the GRASP (as the route building works similarly). Moreover, this difference is further stressed in the last six instances, which have more possible depot locations. Thus, as the TS-GRASP has a better location algorithm, it does not waste so much time trying to obtain the best depot configuration, resulting in smaller computing times and allowing further intensification of the routing phase (enabling to find better results).

3.3 Summary

In this chapter two specific problems in the LRP literature were addressed: the CLRP and the LARP. Both were formally defined, existing approaches reviewed and new (heuristic) approaches proposed.

Regarding the CLRP, the new metaheuristic (AGS) is composed of two phases, in which GLS is used to build the starting solution, embedded in a hybrid extended savings algorithm (in the first phase), as well as to control a reduced composite local search (in the second phase). The algorithm was tested on three sets of instances from the literature (a total of 83 different problems) with up to 318 clients, and a comparative analysis (with other published approaches) was performed. Results suggest that AGS is competitive, providing best average results for two out of the three sets with reasonable computing times. Moreover, new best results were found as well as several best known results were matched.

For the LARP, several new approaches were proposed and analysed (two constructive methods, three heuristic improvements and, using these, four metaheuristic approaches). Due to the absence of benchmark instances in the literature, a new set of instances was devised, deriving from the CARP literature. The TS-GRASP outperformed the remaining approaches, results wise, and was extremely competitive regarding computing times. Moreover, the computational analysis allowed to validate the proposed instances as they appear to be balanced (regarding location and routing costs) and representative of several different cost configurations.

Chapter 4

Location-Routing of Semi-Obnoxious Facilities

The previous chapters addressed location-routing approaches, firstly, reviewing the different problems in the literature, afterwards, proposing approaches for two basic single-objective models. Here, the location-routing of semi-obnoxious facilities is mainly studied. These are facilities which combine both desirable and undesirable features, and, as such, should be addressed with multi-objective approaches.

As the focus of this chapter is the semi-obnoxious facility location-routing problem (LRP), firstly, a review (and term definition) of facility location models dealing with undesirable facilities is made, where, models (and objectives) used in the location literature are introduced. Then, the integrated location-routing approach is reviewed, focusing on the main modelling issues. Finally, a multi-objective capacitated LRP (CLRP) is introduced, formally defined, and solved using a newly developed evolutionary algorithm.

4.1 A Review of Obnoxious and Semi-Obnoxious Facility Location

The location of undesirable facilities has been the subject of increased concern as environmental aspects and quality of life are becoming increasingly important in modern societies. Despite the fact that these facilities are, in general, necessary to communities, their location might cause a certain disagreement; which has become an opposition toward the installation of undesirable facilities close to people. Recently, a new nomenclature has been developed to define this opposition (Colebrook and Sicilia, 2007): NIMBY (not in my back yard), NIMNBY (not in my neighbour's back yard), NIABY (not in anyone's back yard), NOPE (not on planet earth), and BANANA (build absolutely nothing anywhere near anyone). Also, several terms, such as "obnoxious", "semi-obnoxious", and "undesirable" have been used (although often wrongly made indistinctively) to classify different types of facilities.

Obnoxious facilities are potentially dangerous (such as nuclear or highly toxic chemical plants) due to, in the event of an accident, being able to cause serious problems to safety or health which would be felt over a wide area and possibly for a long time (Erkut and Neuman, 1989).

Semi-obnoxious is addressed by Church and Garfinkel (1978) to name facilities often indispensable to communities however, due to the nature of the services they provide, causing an undesirable or even harmful effect to communities when installed in its proximity (e.g. landfills, prisons, fire stations, power plants, airports and hospitals). Another term used for semi-obnoxious facilities is “semi-desirable” (both have been recently used interchangeably).

The term “undesirable” is often used undifferentiated to address the previous terms. An undesirable facility causes nuisance or deteriorates, in some measure, the surrounding environment (Melachrinoudis and Cullinane, 1986). This terminology will also be used here, where, undesirable facilities can be either obnoxious or semi-obnoxious.

The risk these facilities pose and the nuisance to nearby communities has led to a differentiated analysis when addressing their location. In fact, these are the two main aspects in the location of undesirable facilities: risk (to both the environment and communities), in the event an accident occurs (often related with an occurrence probability) causing serious consequences to the surroundings; and nuisance or obnoxious effect (perceived by communities), which has a continuous and lasting effect.

The first study addressing undesirable facilities is by Church and Garfinkel (1978) which investigated the maxisum location problem, equivalent to the median problem, but using a maxisum objective instead. It intends to locate a facility that maximizes the sum of distances between the facility and all of the demand nodes (e.g. population centres). These became known as obnoxious facility location problems.

Another way to model the location of undesirable facilities consists of adding to a minimization problem, lower bounds on the distance between clients and facilities (a forbidden zone imposed by the clients). These are usually semi-obnoxious facility location problems, as the facility provides a desirable and needed service, but is not required to exist too close (within the forbidden zone) to a given population centre (Moon and Chaudhry, 1984). Additionally, it can be devised scenarios where from a facility point of view it is desired to be located as far as possible from some clients, as these have an undesirable effect (e.g. locating in areas with high taxes, rents or criminality).

In this work both problems will be reviewed (as they share the same objectives), however, the focus will be on the semi-obnoxious case. An example would be, when addressing the location of an airport. On one hand, customers would like the airport to be close so that they do not need to travel a long distance to receive service. On the other hand, customers do not want the airport to be too close because it generates noise and pollution (Berman and Wang, 2008).

As follows, a brief review will be made on used models and approaches for both these problems according to: firstly, the number of objectives; and secondly, the number of facilities to install. The solution space will also be addressed, which can be discrete, network or continuous. More comprehensive reviews can be found in Erkut and Neuman (1989), Cappanera (1999), Plastria (1996), Ferreira (1997), Carrizosa and Plastria (1999), and Krarup et al. (2002); the latter three specifically addressing the semi-obnoxious case.

4.1.1 Single-Objective Approaches

When locating p undesirable facilities, most models handle a single-objective, in the simplest case, handling the location of a single facility ($p = 1$). When more than one facility has to be located ($p > 1$), not only the location has to be considered but also the allocation of clients to facilities. Both cases are addressed hereafter, where the majority regard obnoxious facilities.

Single-Facility

In single-facility location models only the distance between the facility to install and the clients is relevant, thus, criteria are usually maxisum or maximin to model, respectively, the undesirable median problem and the undesirable centre problem.

The work of Church and Garfinkel (1978) addressed the location of an obnoxious facility on a network with an exact algorithm, where clients are on the vertices of the network and the facility can be located in any of the vertices or in any point in the edges. The objective function consisted in maximizing the weighted sum of the distances between the communities and the facility (1-maxisum). These problems, on networks, are also known as the 1-maxian or anti-median as the objective is to ensure clients are as far as possible from facilities, hence, replacing the median minimization objective, used in desirable facilities, with a maximization objective.

Minieka (1983) addresses both the anti-median and the anti-centre problems on a network. The anti-centre, similarly to the anti-median, seeks to minimize rather than maximize the objective of the desirable facility location counterpart (in this case, the 1-centre problem).

The 1-maxisum problem in the plane is addressed by Hansen et al. (1981) and Hansen et al. (1985), in the first with a minisum objective (using a decreasing non-linear function of distance). It is proposed a geometrical solution technique: big square small square (BSSS). The BSSS method is a continuous branch-and-bound procedure which starts with a square containing the admissible region. The square is continuously divided into smaller squares, discarding those found to be unfeasible or non promising. The procedure stops when the side of the square is inferior to a predetermined value. The method was later extended by Plastria (1992) to the general single facility location problem in the plane.

Maximin models were proposed by Dasarathy and White (1980) and Drezner and Wesolowsky (1980). The unweighted maximin model is solved by Dasarathy and White (1980) in a convex polyhedron in $\mathbb{R}^d$, using Euclidean distances, and suggest enumerating all local optima using the Karush-Kuhn-Tucker (KKT) conditions. Drezner and Wesolowsky (1980) deal with the maximin model on the plane, with weighted Euclidean distances to reflect the relative importance of existing facilities, and propose a numerical bisection search procedure. Both methods suggest simple graphical solutions involving growing circles around facilities or client locations.

The objective in the 1-maximin problem in the plane is to locate a facility so as to maximize the distance to any of the clients. Cappanera (1999) observes this is equivalent to the geographical problem of finding the largest circle not including any client, the centre of it, being the optimal location of the undesirable facility. On a network, the unweighted 1-maximin problem is trivial (an optimal location always exists halfway the longest edge), and the weighted problem, according to Erkut and Neuman (1989), can be solved by considering all pairs of vertices (polynomial time algorithms can be found in Melachrinoudis and Zhang, 1999, Berman and Drezner, 2000, and Colebrook et al., 2002, being the only works found to address this problem on a network).

Melachrinoudis and Cullinane (1985) present a formulation for the 1-maximin problem constraining the location of the facility to a bounded region, and additionally assuming a minimum circular region around the clients where location is not allowed. Karkazis and Karagiorgis (1986, 1988) present exact approaches equally for the 1-maximin on the plane; in the latter work with the use of protected or forbidden regions (not necessarily circular) where locating the facility is not allowed, for example, nature reserves or parks.

Melachrinoudis and Cullinane (1986) consider a more realistic formulation, where the maximum weighted inverse square distance between the facility and the clients is minimized (1-minimax): $1/e^2$, where e is the distance to the facility. This is due to the various types of pollution, such as radiation and noise, which tend to decrease according to the proposed measure.

A version of the weighted 1-maximin problem in a convex polygon was studied by Erkut and Öncü (1991). The weights are functions of a parameter, which by using different values can obtain different disutility (obnoxious) functions for the clients. The authors show this problem to be equivalent to the 1-minimax problem where the costs are a decreasing function of distance.

Melachrinoudis and Smith (1995) developed an algorithm for the Euclidean weighted maximin problem in a polygonal region using the weighted Voronoi diagram as a data structure.

Erkut and Neuman (1989) observe that the maximin objective (in the single-objective case) is better suited than the maxisum objective for determining the location of undesirable facilities. This is due to: the first aiming at obtaining the individual biggest minimum distances; the second concerning the sum of distances, which may result in the optimal location in the vicinity of clients. This can be overcome by adding constraints imposing a lower bound on the distance to the facility. Still, the maxisum objective is useful when solving multi-objective problems.

Muñoz-Pérez and Saameño-Rodríguez (1999) develop the problem of locating an obnoxious facility, using Euclidean distances, in a bounded polygonal region which includes several existing facilities (or population centres). The region may contain forbidden polygonal regions. The proposed general problem can lead to, among others, the maxisum and the maximin problem. The authors show that an optimal solution can be found in polynomial time, and such an algorithm is presented in Saameño-Rodríguez et al. (2006).

Besides the usual maxisum and maximin objectives, minisum objectives have also been studied. When using minisum objectives, two situations are usually considered: the facility is obnoxious and a measure other than distance is used (inversely proportional to distance) (Hansen et al., 1981; Sung and Joo, 1994); or, it is intended to minimize another aspect (e.g. transportation cost) and the obnoxious effect is handled by a lower bound on the distance to the facility (Brimberg and Wesolowsky, 1995; Berman et al., 2003). The latter approach is more suitable for the location of semi-obnoxious facilities.

Sung and Joo (1994) present an exact solution technique for locating an obnoxious facility on a network, where damage could be inflicted within a distance from the facility. The objective is to find a location minimizing the sum of weights within the circle centred at the location point (1-minisum). Each weight is assigned to a point on the network and represents a numerical scale reflecting the extent of undesirability resulting from the proximity to the facility.

The 1-minisum on the plane was studied by Brimberg and Wesolowsky (1995), where the objective function measures the transportation cost, while social (obnoxious) costs are included implicitly as distance lower bound constraints. The authors propose a branch-and-bound algorithm for rectilinear distances.

Other objectives and issues have also been proposed recently.

Gordillo et al. (2006) handle the location of a semi-obnoxious facility in the plane where clients are required to be serviced and a set of populated areas (defined using convex polygons) is to be protected from the obnoxious effect of the facility. The problem if formulated as a margin minimization model, and a polynomial solution method is presented.

Berman et al. (2003) and Berman and Wang (2008) study the location of semi-obnoxious facilities where, if the location is too close to clients, expropriation is allowed at a given cost. In both works the problem is formulated on a network. In the first using: a maximin model considering the distance from the facility to the non-expropriated nodes, subject to an expropriation budget; and a minisum of the expropriation costs, while ensuring the facility is not within a given distance (related to the maximal covering location problem, of the desirable location literature). In the second: the difference between the maximum and the minimum weighted distances is minimized; and the minimum weighted distance is maximized, subject to a limit on the maximum weighted distance.

Multi-Facility

Addressing the location of several facilities requires to consider not only the distances between facilities and clients but also between facilities. Both will be analysed here, however, while the first are clearly directed at addressing the location of undesirable facilities, the second are mostly directed at dispersion models (as the obnoxious effect is among facilities), in which they are to be located in such a way as to affect each other the least possible.

According to Ferreira (1997), objectives are usually: maximinmin, maximinsum or maxisummin. These objectives can be analysed based on the operator found in the three syllables which compose them (Erkut and Neuman, 1989). For example, maximinsum has, respectively, the operators: “max”, “min”, and “sum”.

When the operator in the second syllable is “min” (maximinmin and maximinsum), it is sought an equitable solution, as it is maximized the performance (minimization of the obnoxious effect) in the worst case for each location. If it is “sum” (maxisummin), it is maximized the overall systems performance, possibly at the expense of individual results.

The operator in the third syllable determines the interactions most valued by the model. When "min" (maximinmin and maxisummin), the model considers only the minimum distances between the several entities, for example, between facilities and clients. Using the operator "sum" (maximinsum), the model handles the sum of the distances between the entities.

Looking at a specific example: the maximinsum model aims to maximize (1st operator) the minimum (2nd operator) of the summed (3rd operator) distances between, for example, facilities.

For the location of p obnoxious facilities in the continuous space, Drezner and Wesolowsky (1985) propose two formulations: p-minimaxsum, where it is minimized the maxisum weighted distance between facilities and demand points, subject to lower bound constraints on the distance between them; and p-maximinsum, where it is maximized the minimum weighted distances between the facility and the demand points, subject to upper bound constraints on the distance between them. The authors show that both problems are linked by duality and present an exact approach for location on $\mathbb{R}^1$ (i.e. a line). Giannikos (1993) also proposes formulations for location in the plane. If the demand point (client) is only affected by the closest facility, the use of p-maximinmin models is suggested; if the client is affected by all the facilities, a p-maximinsum model should be used.

The location of several facilities on networks where it is intended to maximize the sum of minimum distances between facilities and clients (p-maxisummin model) is known as the p-maxian or anti-p-median problem (Church and Garfinkel, 1978). Without any further constraints all p facilities would be located nearby (which may prove useful when defining zones for polluting factories, having them aggregated rather than dispersed). This problem was addressed by Erkut et al. (1990), where it is shown to be NP-hard, and an exact branch-and-bound method and a heuristic procedure are presented. The same problem on a tree has been addressed by Burkard et al. (2007), where it has been shown that it can be solved in linear time.

The discrete anti-p-centre problem (extension of the anti-centre to the multi-facility case), introduced by Klein and Kincaid (1994), aims at maximizing the minimum weighted distance between demand nodes and their nearest facility (belonging to the class of p-maximinmin models). The authors present an algorithm to solve this problem in polynomial time. Later, Zmazek and Žerovnik (2004) presented an algorithm with linear time complexity for this problem.

Moon and Chaudhry (1984) address the network location of facilities requiring to be separated from, either other facilities, or demand points (using distance constraints). p-maximinmin models are used and, when it is considered the inter-facilities distances (intended to ensure dispersion of facilities throughout the network; hence clients not playing any role in the model) the problem is called anti-covering concerning the location of the maximum number of facilities so that no two facilities are within a given time or distance of each other.

The anti-covering problem on networks (similar to the maximum independent set problem) was also addressed by Chaudhry et al. (1986), Murray and Church (1997), and Chaudhry (2006) where, respectively, a greedy heuristic, a Lagrangian relaxation solution approach, and a genetic algorithm are used. New formulations are provided by Erkut et al. (1996) where linear relaxations provide, in most cases, 0-1 solutions to the problem. Tamir (1991) presents a linear time algorithm for this problem on trees. This problem is also addressed as the *r*-separation problem, with *r* being the minimum value allowed between two points (facilities).

If rather than constraining the minimum distance between facilities, it is intended to maximize it, the problem becomes a *p*-dispersion problem, also known as the maximum diversity problem (seeking for the maximum dispersion between facilities, e.g., locating mutually undesirable facilities such as: missile silos, that pose a threat to each other; franchises, that when located too close together can affect each other's profits; radio transceivers to service cellular phones, in order to minimize interference problems). This is the most used model in locating undesirable facilities mainly due to its duality with the (p − 1)-centre problem (from the desirable facility location literature) on specific networks: trees (Shier, 1977). For the *p*-dispersion problem on trees, Chandrasekaran and Daughety (1981) presents a polynomial algorithm. On general networks, this problem is proven to be NP-hard (Erkut, 1990), hence, several approaches have appeared in the literature. Erkut (1990) presents both an exact branch-and-bound method and a heuristic (combining a greedy method with a 2-opt local search). Several other heuristic methods such as simulated annealing (Kincaid, 1992), tabu search (Kincaid, 1992; Palubeckis, 2007), and greedy randomized adaptive search procedure (Ghosh, 1996; Resende et al., 2010) have been proposed. Also, other variants of the problem have appeared, like the *p*-dispersion under facility capacity and budget constraints (Rosenkrantz et al., 2000).

On the plane, the *p*-dispersion problem has been addressed by Drezner and Erkut (1995) where, the relationship with the *p*-circle packing problem is investigated (the *p*-dispersion in a square is equivalent to packing *p* circles with maximal radius), and a non-linear programming formulation is presented.

Erkut (1990) also proposes a model similar to the *p*-dispersion where, rather than considering only inter-facility distances, it is also considered facility-to-client distances. This problem is named *p*-anticentre-dispersion and is proven to be NP-hard on general networks (Tamir, 1991). The version of this problem on the plane is presented by Brimberg and Mehrez (1994) and, the $p = 2$ case is later addressed by Tamir (2006) where subquadratic algorithms are proposed.

If rather that maximizing the minimum distance between facilities, it is intended to maximize the smallest sum of distances (consider the overall obnoxious effect, rather than the closest individual one), we are dealing with the *p*-dispersion-sum problem, which belongs to the class of *p*-maximinsum models. This problem on networks was addressed by Kincaid and Yellin (1993) where: polynomial algorithms are found for the location of less than four facilities on trees; and, a simulated annealing metaheuristic is presented for general networks (as the problem in this case is NP-hard). Pisinger (2006) studies the same problem on networks, presenting an exact branch-and-bound algorithm (solving moderately large problems to optimality) and an heuristic based on Lagrangian relaxation, concluding that, although the *p*-dispersion-sum problem is difficult to approximate in the worst-case situation, greedy heuristics may perform very well on instances occurring in practice.

Another way to promote facility dispersion is given by the *p*-defense problem. This problem is a *p*-maxisummin model in which it is sought to maximally disperse facilities through the use of the sum of the minimum distances between them. Moon and Chaudhry (1984) identifies this problem, which has several applications in military defense, where it is common to scatter installations in order to make it more difficult for enemies to disarm them (justifying it's name).

The only problem found to explicitly address the semi-obnoxious p facilities location was presented by Carrizosa and Conde (2002), where facilities are to be located on a network maximizing the average distance to population centres (distributed in the plane) per unit of transportation cost (function of the network distances). The authors aggregate into a single objective function transportation cost (to be minimized) and the sum of individual environmental utilities (to be maximized). To tackle this, they propose a fractional objective representing the environmental utility per transportation cost (to be maximized).

4.1.2 Multi-Objective Approaches

Generally, in multi-objective approaches, in order to consider the location of undesirables facilities three different types of criteria are used (usually in a bi-objective model): cost, risk/obnoxious effect, and equity. Also, while on the single-objective approaches most works consider locating obnoxious facilities, where risk/obnoxious effect is usually taken into consideration (when additional issues are concerned they are in the form of constraints); on multi-objective approaches, semi-obnoxious facilities are mostly addressed, as these are prone to be tackled with multi-objective models (due to its attracting and repelling effects).

As follows, and due to the smaller number of works concerning more than one objective (the review of Erkut and Neuman, 1989, points to a lack of previous works), both single and multi-facility location will be simultaneously addressed. For concepts regarding multi-objective problems the reader is referred to Section 5.1.1.

One of the first works in this subject is due to Ratick and White (1988) where a model for locating obnoxious facilities with three objective functions is studied. Two new concepts are added to the traditional (desirable facility) location approach (intended to locate close to population centres in order to reduce cost): perceived risk (by the communities), which is a function of the size of the facilities and opposition faced increases disproportionately (i.e. if a facility is twice the size of another, the public will perceive the resulting burden, or disutility, as being more than twice); and a measure of equity, called a "complementary anticover", measured as a function of the number of installed facilities. Based on these objectives, the authors propose the location of smaller facilities (rather than a single large one), although higher costs are incurred.

This work laid the foundation to the multi-objective approach by Erkut and Neuman (1992) addressing a three-objective (mixed-integer) model on networks in which trade-offs are made between: cost, regarding both location and transportation; opposition, resulting from the sum of the individual disutilities (hence proportional to the population of the community) which, regarding each facility, is a non-linear decreasing function of the Euclidean distance to it (following the measure proposed by Melachrinoudis and Cullinane, 1986, but using a parameter as the inverse power of distance), and a non-linear increasing function of the facility size; and equity concerns, where the maximum individual disutility is minimized. The final two objectives follow the concepts addressed in single-objective approaches, where respectively, minisum and minimax objectives are used. The authors also point as a future research the incorporation of transportation risk into the model (suitable when dealing with hazardous materials – HAZMAT), thus advocating the need to address both location and routing in a combined model.

Melachrinoudis et al. (1995) study the location of semi-obnoxious facilities (landfills) using a dynamic (multi-period) mixed-integer programming model. They propose four objectives, which minimize (over the planning horizon): cost, composing the sum of location and transportation costs; total risk, posed to all human population centres; total risk, posed to all non-human population centres; and disequity, a measure of the maximum risk on each individual over all population centres. Discrete location is addressed, as is assumed a previous analysis narrowing down the number of potential locations. Unlike previous model, not only human but also non-human population centres (ecosystems) are considered. Cost minimization is found to be in conflict with the other three risk related, and positively correlated, objectives (which is to be expected as risk minimization is achieved by opening and shipping to the most remote landfills, in turn increasing transportation cost). Since on the test example the proposed model generated 100 efficient solution alternatives (far exceeding the maximum limit of nine, pointed by the literature, Miller, 1956, as the limit for an effective evaluation by the decision maker – DM) a filtering method is used to reduce the number of alternatives to be presented to the DM. To be noted that a high number of efficient solutions is common for multi-objective models.

The formerly presented works are the only to study the location of undesirable facilities with more than two objectives (even though some are highly related). The remainder of the reviewed papers focus on bi-objective models, commonly, with opposing push (risk/obnoxious effect minimization, typically using distance maximization as a surrogate) and pull (cost/distance minimization) objectives.

The location of an obnoxious facility on a general network was addressed by Zhang and Melachrinoudis (2001), where a bi-objective model maximizes the sum of weighted distances between the facility and the vertices (maxisum) and maximizes the minimum weighted distance from the facility to the vertices in the network (maximin). A polynomial algorithm is developed in order to generate the efficient set.

When dealing with semi-obnoxious location a model combining push and pull objectives is most suited. This has led Hamacher et al. (2002) to study models with maxisum (push) as well as minisum (pull) objectives. The authors present a polynomial time algorithm (using CPLEX as a solver) and generalize the results to also incorporate maximin (push) and minimax (pull) objective functions.

Also on networks, Colebrook and Sicilia (2007) handle the location of undesirable facilities with the λ-anti-cent-dian problem. This problem analyses the undesirable centre and median models which, on the location of desirable facilities, is addressed as the λ-cent-dian problem (Halpern, 1978), combining the minimax (centre) and minisum (median) objectives by a parameter λ. On the undesirable facility location the objectives used in the convex combination are the maximin and maxisum. The authors present a polynomial algorithm (with the same complexity as the work by Hamacher et al., 2002).

On the plane, all papers handle the location of a single semi-obnoxious facility (the only exception being the paper by Rakas et al., 2004, where several of these facilities are to be located). These will be discussed as follows.

Romero-Morales et al. (1997) use two objective functions: one measuring the environmental impact caused in communities by the facility (with a non-increasing convex function of the distances); the other, gathering the transportation costs between the facility and the communities (using a Lipschitz-continuous non-decreasing function of distances). They propose an approach based on the BSSS method, with a new bounding scheme which exploits the structure of the problem based on Lagrangian relaxation techniques.

The objectives used by Brimberg and Juel (1998) are: minimization of the weighted sum of the Euclidean distances between the facility and the communities (intended to measure the transportation cost); and the minimization of the weighted sum of the Euclidean distances between the facility and the communities, raised to a negative power (in an attempt to estimate social rejection or environmental impact resulting from the installation of the facility). The approach considers the convex combination of the two minisum objective functions, varying the weights attributed to each objective function in order to obtain a trajectory of the efficient set. The existence of a discontinuity in the trajectory makes more difficult the process of obtaining solutions. The same push (depending positively on the distance function) and pull (depending negatively on the distance function) objectives were addressed by Skriver and Andersen (2003) (instead of combining the two objective functions, the authors considered the model bi-objective) where an adaptation of the BSSS method tackles the problem on the plane and on networks.

Melachrinoudis (1999) presents the problem of locating a semi-obnoxious facility in the plane where other facilities already exist. It is intended that the new facility be installed near existing ones in order to minimize transportation costs and, at the same time, due to its nuisance, farthest possible. Thus, minisum and maximin objectives are considered. As the number of constraints in each of linear bi-criteria problems is small, the author solves them using an adaptation of the Fourier-Motzkin elimination method. This method, in each iteration, reduces in one the number of variables but increases the number of constraints, thus only suited for problems with few constraints. The same objectives are addressed by Melachrinoudis and Xanthopulos (2003) where an algorithm is developed to obtain the complete set of efficient solutions. They solve the discontinuity problem in the set of efficient solutions raised in Brimberg and Juel (1998) and found that most of these efficient solutions are on the edges of a Voronoi diagram.

Algorithms for minisum-maximin models have also been presented by Blanquero and Carrizosa (2002) and Karasakal and Nadirler (2008). In the former, the authors present an algorithm to obtain a finite feasible subset that approximates the Pareto-optimal front for the bi-objective problem. In the latter, a three-phase interactive geometrical branch-and-bound algorithm is suggested to find the most preferred efficient solution. The first two phases aim at eliminating the parts of the feasible region to which inefficiency is proved. The third phase is based on an interactive search in the remaining regions with the involvement of a DM where she/he is given the opportunity to use either an exact or an approximate procedure to carry out the search.

Ohsawa and Tamura (2003) extend the model by Ohsawa (2000) (which uses maximin and minimax criteria) to address elliptic maximin and rectangular minisum criteria using different metrics simultaneously; in what the authors consider to be a more suitable approach for analyzing real-world location decisions. Polynomial time algorithms for finding the efficient set and the trade-off curve are presented in both works.

A bi-criteria model seeking the lowest effect on the population at the highest level of protection is presented in Plastria and Carrizosa (1999). This is done by taking into account a radius of influence to be maximized (indicating within which distance from the facility, population disturbance is taken into consideration) and the total covered population (within the influence radius from the facility) to be minimized. They developed low complexity polynomial algorithms to construct the complete trade-off curve between both objectives together with corresponding efficient solutions.

Yapicioglu et al. (2007) introduce a model composed of a weighted minisum function (for transportation costs) and a distance-based piecewise function (for the obnoxious effects of the facility). A bi-objective particle swarm optimizer is devised to produce a diverse set of non-dominated solutions. An analysis on the method's computational complexity shows a linear increase in effort with problem size.

The only work to address the location of multiple facilities is by Rakas et al. (2004) with a bi-objective model minimizing costs (concerning both transportation and installation) and political opposition by communities (obtained with a newly devised scenario-specific formulae). The authors reduce the number of potential locations using multi-attribute decision methods and apply the developed approach to a real case in Maryland, USA. Then, CPLEX is used to obtain the optimal solution (solving a weighted sum of both initial objectives). A method is also proposed to address uncertainty (on the amount of garbage generated per area, as well as the cost) using fuzzy linear programming.

4.2 A Review of Obnoxious and Semi-Obnoxious Location-Routing

When routing is considered in the location of undesirable facilities, two scenarios can be devised: when the transported product is undesirable (due to causing risk or nuisance) and the corresponding transportation has to be treated accordingly; and, when no risk/nuisance exists in transportation (albeit the facility where the product is stored is undesirable itself).

When the transported product is undesirable it typically refers to waste, where one can classify it into high-, medium-, and low-level waste (Boffey et al., 2008). High-level waste (also known as HAZMAT) could, in case an accident occurs, cause serious problems which would be felt over a wide area and possibly for a long time (e.g. nuclear or highly toxic chemical waste). Moreover, the disposal facilities which handle it are clearly obnoxious and should be located far from population centres. Low-level waste (e.g. domestic or non-toxic industrial waste) on the other hand, is more of a nuisance than a danger, and its effect (noise, smell, vermin, insect spread disease, etc.) is relatively restricted. The effect of a single shipment may be short lasting, but the overall effect of repeated shipments can be long lasting. Moreover, the effect of the corresponding disposal facilities may be mild and, when contemplated, it may be done by merely imposing a lower bound on the distance to the facility. Medium-level waste has characteristics of both high- and low-level waste.

In the low-level waste case, although the transported materials have undesirable features, the facility where they are transported to, often has not. Even though these may not be regarded as (semi-)obnoxious facilities, works handling their location are discussed here for the sake of completeness. On the reviewed works, none addresses the location of undesirable facilities with non-undesirable products. Hence, in most cases, rather than considering routes, in order to minimize transportation risk/nuisance, paths in-between the demand and supply points are used. The location to be determined is usually between these points.

A survey of works addressing only routing of HAZMAT is given by Erkut and Verter (1995). Regarding problems in this area handling both location and routing, several authors point to the lack of works (Boffey and Karkazis, 1995; Lozano and Mesa, 2000). This view is confirmed in the previously presented taxonomy (Chapter 2) where few works handle multiple objectives and even less the location of obnoxious or semi-obnoxious facilities (inherently multi-objective). Here, existing works will be analysed (both formulation and algorithms wise), where only the work by Boffey et al. (2008) handles continuous location.

For the single-objective case, only two papers handle the location-routing of undesirable facilities:

- The paper by Cappanera et al. (2004), where a model is presented for the location of obnoxious facilities and routing of HAZMAT between facilities and communities. The objective is the minimization of the totals costs composed of installation and transportation (between the points where materials are produced and installed facilities) costs; establishing upper bounds on the communities exposure to the obnoxious effect resulting from both location and transportation. They consider a directed graph and present two Lagrangian heuristics and a branch-and-bound algorithm for this problem. Computational results are shown on randomly generated data.
- The work by Nema and Gupta (1999), presenting a model where both total costs (location and transportation) and total risk (site and transportation) are weighted and combined into a single utility function. Two different types of facilities are to be installed (treatment and disposal sites), and an example network is addressed consisting of 16 nodes (six being waste generators, two representing potential treatment sites and another two potential disposal sites) and 20 links. The example is solved to optimality using five different combinations of weights in the utility function.

The remaining works address several objectives, where, following the location literature, can be categorized as: cost, risk/obnoxious effect, and equity. Cost is usually based on network distances (or a surrogate as time) for transportation and facility installation costs; while risk is often obtained by measuring the population exposure to the HAZMAT (during either transportation or processing/storage/disposal at facilities). In Chapter 2, an overview of the objectives used in each of the following papers is shown. Moreover, high-level waste has received the most attention, being addressed firstly.

Zografos and Samara (1989) present a model addressing minimization of: travel time, transportation risk, and disposal (location) risk. The travel time is associated with the links of the transportation network. Transportation risk is defined as the product of the probability of an accident occurring with the consequence of that accident. For measuring disposal risk, the total distance between population centres and the disposal site is used as a surrogate (the greater the total distance, the lower the risk), seen as a maxisum objective (similar to the ones used in the location literature). The work is mostly theoretical and can be seen as the first integrated location-routing approach handling HAZMAT. The authors propose as future research further data collection for calculating routing risk, in which can be seen as one of the biggest drawbacks of this model, as it may be difficult to estimate.

In the work by List and Mirchandani (1991) cost, risk and risk equity are considered. Costs are related with network distances, while risk are considered "zonal" attributes as their effect is spread over the plane. The zonal risk is considered as the sum of the risk resulting from transportation and processing, storage or disposal at facilities. Risk equity is measured as the maximum zonal risk per unit population (to be minimized). The authors present a model and apply it to a case study in Albany, New York, where trade-offs for each objective are analysed.

Minimization of transportation burden and perceived risk are addressed in the model by ReVelle et al. (1991). Albeit the material is hazardous, the facility is not considered so (as is considered that in case an accident occurs, impacts are confined to the facility itself, due to the presence of containment/cleanup personnel, therefore, the general public is not threatened or even aware of these accidents), hence, the risk only concerns the transportation activity. The transportation burden is measure in ton-miles (weight carried per distance, a surrogate for transport cost) and the perceived risk in tons-past-people. Here, the risk is not considered as a probability of shipping accidents happening (as in previous models) but as the people's perception of risk, which depends on the quantity carried past population centres (population exposure). Optimal solutions are found for the presented case (by minimizing the convex combination of both objectives) and the authors demonstrate how transportation burden and risk influence location decisions.

Boffey and Karkazis (1993) review existing HAZMAT models for separate location and routing, as well as the integrated location-routing approach. General models are proposed for LRPs concerning HAZMAT where the facility may or may not be obnoxious.

The location-routing of a single obnoxious facility is considered by Stowers and Palekar (1993). A bi-objective minisum-minimax (of total transportation and location risk) model is developed, quantifying the total exposure of the population during transportation as well as long term storage (location). Although only addressing risk in their model, the authors show that for the class of bi-objective location problems considering only travel risk and travel cost, an optimal solution is at a node, thus suggesting the problem only becomes difficult when location risk is considered. When location risks are considered and population is concentrated at nodes, the problem exhibits a finite dominating set.

Jacobs and Warmerdam (1994) model the hazardous waste LRP as a continuous network flow problem. The proposed model minimizes a linear combination of risk and cost in time. Risk is defined as the total probability of an accident occurring during transportation, storage or disposal.

To demonstrate the model, the authors describe a hypothetical location and transportation network for a 10-month planning period.

Current and Ratick (1995) propose a model where both materials and disposal sites have a risk associated (HAZMAT and obnoxious facility). One of the objectives is cost minimization, comprising of per unit transportation (network) costs and fixed and variable costs at facilities. Risks and equity are spatially determined (risk is measured with population exposure) and are addressed on an aggregated (using two minisum objectives) and individual level (using two minimax objectives) for both transportation and location. The model is mixed-integer (with binary variables only for location decisions) totalling five objective functions, and is solved to optimality in a sample network with 50 nodes (15 waste generating sites, 31 transportation nodes and four potential location sites) and 146 directed arcs. The authors point the main problem not being solving the problem, but rather generating all the efficient solutions and interpreting the trade-offs between them, classifying it as a “daunting if not intractable task”. Hence, they propose the use of interactive decision support systems in order to reduce the total number of solutions to be generated and analysed (and possibly facilitate the analysis).

A similar formulation is provided by Wyman and Kuby (1995) but considering risk and equity for transportation and location jointly (thus providing only three objective functions). Additionally, the selection of different technologies for treatment facilities is considered, where a new solar-driven waste detoxification is compared with incineration regarding cost, risk, and equity. Model wise, new risk and (dis)equity measures are proposed. For risk, transportation and location (treatment) risks are measured in population, times the concentration to which they are exposed, multiplied by the number of hours they are exposed ($\mu g/m^3$ person hrs). For equity, a minimaxsum objective is used (minimizing the maximum sum of kg*km shipped to facilities), based on the premise that is most equitable to treat waste where is produced. This formulation leads to similar results as Current and Ratick (1995), but rather than seeking to equitably share risks, it is attempted not to take on other people’s risks (assuming risks should be incurred by the people who produce waste).

Giannikos (1998) considers four objective functions for discrete location of garbage treatment facilities, in which garbage (deemed obnoxious) has to be transported through a network representing the road between possible facility locations and communities. The author presents an approach using goal programming in which the objectives are: total cost (transportation and location), minimization of total perceived risk (similar to population exposure), equitable risk distribution in the transportation activity, and equitable distribution of the disutility (obnoxious effect) caused by facilities. A theoretical example using 13 nodes, in which 3 are communities that produce garbage and 5 are possible facility locations, is used to test the approach. Functions of growing penalties were considered, allowing better results, since large detours of the goals were more penalized. Difficulties of this model come from: the correct selection of the priority weight expressing the relative importance of each goal; the goals to achieve; and the selection of the type of penalties to establish to each objective.

A bi-objective model, minimizing total cost and transportation risk, is proposed by Alumur and Kara (2007). As in the work by ReVelle et al. (1991) the facility is not deemed obnoxious.

Additionally, the authors consider several technologies (different types of facilities) and define a set of constraints to ensure that a given treatment technology (type of facility) is opened only if a minimum amount of materials for that technology exists. A large-scale implementation of the model in the Central Anatolian region of Turkey is presented (with 92 generation nodes and with 15 and 20 candidate sites). The problem is solved to optimality using CPLEX, although the authors point the need to develop heuristics (even more as none was yet existing) in order to tackle larger problems.

More recent approaches have addressed the location-routing of low-level waste.

The work by Boffey et al. (2008) studies such problems using four of the five objectives provided by Current and Ratick (1995), the only excluded being the one concerning minimizing risk to towns from the facility (handled by imposing a minimum distance constraint between the town and the facility). Also, the potential facility locations form a continuous set, as facilities can be placed on an off-road site. The authors develop a Lagrangian relaxation (heuristic) approach and apply it to a real scenario: the location of a disposal site for low-level waste in the district of Algarve, Portugal. The corresponding transportation network has 945 nodes (from which 84 are towns) and 1218 (undirected) edges.

A heuristic method is also proposed by Caballero et al. (2007) in order to locate two incineration plants in the region of Andalusia, Spain, for the disposal of solid animal (low-level) waste, and design the routes to service the slaughterhouses in the region (unlike the remaining models, rather than paths linking demand-to-supply nodes, routes are used, as products cause nuisance rather than posing risk). Here, both the transported product and the facility are deemed obnoxious, and five objective functions are considered regarding cost (both minisum, one for location related costs and another for transportation costs), nuisance measured as social rejection (both minisum, one regarding the population close to the facility to install and another for the population through which routes pass), and equity (a minimax objective intended to minimize the maximum social rejection of the town most affected by waste transportation). The authors propose a metaheuristic based on tabu search in order to obtain the efficient set as the problem could not be solved using CPLEX. The problem has route capacity and duration constraints, 93 clients to service and 6 possible locations (from which at most two can be chosen).

4.3 A Multi-Objective Capacitated Location-Routing Problem

Previous sections allowed to identify and analyse currently addressed issues in the literature regarding the location(-routing) of undesirable facilities. However, on the reviewed works, none addresses the location-routing of (semi-)obnoxious facilities with desirable products/services, albeit several examples can be thought of (e.g. locating paper mills). Although the location of such facilities raises negative social responses and environmental pollution, routing (of the products) does not, and should be considered in order to correctly manage costs (even if objectives reflect interests of different entities/DMs).

This is a problem which, although it hasn't been specifically addressed in the literature before, can be tackled using some of the existing models. For this specific problem, the CLRP has been extended to address it. This is done by considering several opposing objectives, thus the problem will be named multi-objective CLRP. As follows, a definition will be made for this problem, to which an evolutionary approach is proposed, and corresponding computational analysis presented (Lopes et al., 2010a).

4.3.1 Problem Definition

The multi-objective CLRP can be defined as intended to locate facilities, while drawing the distribution routes of a single product, subject to (homogeneous) vehicle capacity constraints. Unlike the single-objective counterpart where only location and routing cost are considered, several opposing objectives are taken into account. This makes the problem applicable to the location of (semi-)obnoxious facilities, where the distributed product possesses few (e.g. low-level waste) or no undesirable features.

This is the problem addressed here where, as the facility has undesirable features, to the usual cost objective function, two other objectives are considered, namely, facility obnoxious effect and the corresponding equitable distribution.

The considered obnoxious effect, caused by each facility over the communities, depends on the Euclidean distance between the facility and the community and on the size (population) of the community, although an underlying road network is assumed. Following the notation given in Section 3.1.1 for the single-objective CLRP, for each community (client) $i \in I$ and each facility (depot) $j \in J$, the obnoxious effect is obtained by a function that decreases inversely with the square of the Euclidean distance e_{ij} and directly with the size of the community s_i (following the measure by Melachrinoudis and Cullinane, 1986). The system overall obnoxious effect is to be minimized.

In order to equitably distribute the obnoxious effect, for each individual in the community, the overall obnoxious effect is obtained (considering the effect upon the individual of all facilities to be installed), and the maximum of these individual effects is to be minimized.

Thus, for the multi-objective CLRP considered here, the following formulation is used:

$$(\text{CLRP}_2) \quad \min \quad Z^1 = \sum_{j \in J} f_j y_j + \sum_{a \in A} \sum_{k \in K} c_a x_{ak} + \sum_{k \in K} \sum_{a \in \delta^+(J)} F x_{ak} \tag{4.1}$$

$$\min \quad Z^2 = \sum_{j \in J} \sum_{i \in I} s_i \frac{1}{e_{ij}^2} y_j \tag{4.2}$$

$$\min \quad Z^3 = \max_{i \in I} \sum_{j \in J} \frac{1}{e_{ij}^2} y_j \tag{4.3}$$

$$\text{s.t.:} \quad x \in X.$$

The objective function (4.1) is equivalent to (3.1) and manages total costs. Function (4.2) aims at minimizing the total obnoxious effect (a minisum objective), and in order to achieve equity, function (4.3) is used (minimaxsum objective). Constraints used are the same as defined from (3.2) to (3.11).

4.3.2 A Multi-Objective Evolutionary Algorithm

An introduction to multi-objective optimization can be seen in Ehrgott and Gandibleux (2002) where an annotated bibliography and general definitions are given. In order to tackle these problems, a possible approach is to attempt to obtain the full Pareto front using heuristic methods. Evolutionary methods are population based heuristics, which may be more suitable for multi-objective approaches, as it allows to possess several solutions at a given time, possibly approaching the Pareto front more easily. This has led to being one of the most used heuristic approaches in multi-objective optimization. Recent references on multi-objective evolutionary methods are the works by Deb (2001) and Coello Coello et al. (2007).

The evolutionary method chosen in this work was the genetic algorithm, where a general framework, the second version of the non-dominated sorting genetic algorithm (NSGA-II), is adapted and used to tackle the previously defined multi-objective CLRP. The NSGA-II is an elitist algorithm (as it favours the elites of a population) presented by Deb et al. (2002) which has had several successful applications to a wide range of problems (e.g. Lacomme et al., 2006, and Dugardin et al., 2010). The algorithm obtains successive generations of a population of solutions which is partitioned into non-dominated fronts using a sorting method.

The method is called non-dominated sorting and will be described as follows. Firstly, the non-dominated set is obtained and is ranked 1 (forming the front 1) being temporarily disregarded from the population pop. Then, the following non-dominated solutions are obtained, ranked 2 (giving the front 2), and also disregarded from pop. This is performed repeatedly until all solutions are ranked (e.g. Figure 4.1).

After the sorting is performed (with the ranks corresponding to the solution fitness value, being 1 the fittest, hence to be minimized), solutions (named parent solutions) are selected to generate new (children) solutions, at each generation gen, by applying crossover and mutation operators. In order to obtain each parent solution, a crowded tournament selection operator is used. This operator uses binary tournament where, two individuals are randomly chosen in which the fitter (with the smallest rank) of the two is selected as a parent. If the solutions share the same rank, a crowding distance cd is used, from which the solution with the biggest is chosen.

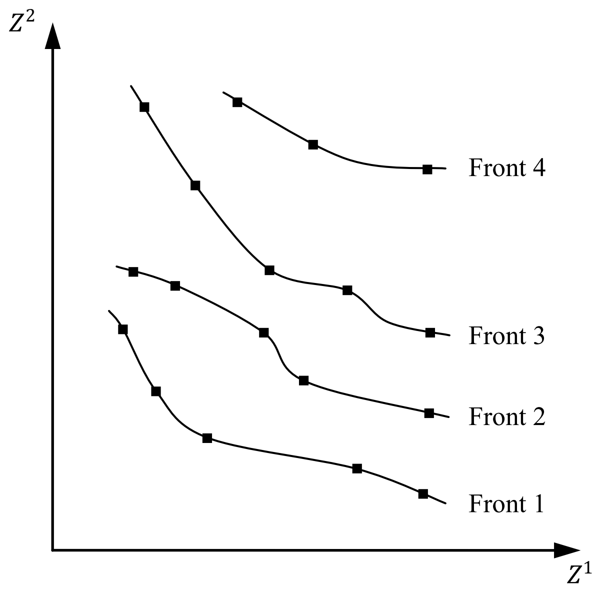


Figure 4.1 Four non-dominated fronts of the population of solutions (minimizing two objectives).

For each objective function Z^c ($c = 1, 2, 3$), corresponding maximum Z^c_{max} and minimum Z^c_{min} values are obtained. Let R be a front of nr solutions, all solutions in R are sorted according to increasing values of Z^c, being R^c_l the lth solution in the sorted front. The corresponding crowding distance, for $1 < l < nr$, is computed as follows:

$$cd = \sum_{c=1}^{3} \frac{Z^c(R^c_{l+1}) - Z^c(R^c_{l-1})}{Z^c_{max} - Z^c_{min}}. \tag{4.4}$$

For $l = 1$ or $l = nr$, $cd = \infty$, and thus corresponding solution is always chosen. This crowding distance favours the extreme points of the front, to try to enlarge it, and promotes the choice of solutions most widely spread in the obtained fronts (thus preventing clustering of solutions). A specific adaptation to the NSGA-II presented here is that, for the crowded tournament selection operator, solutions are randomly obtained from fronts 1 and 2 (unlike usual implementations where the entire population is considered). This allows breeding to be performed between higher quality solutions (as diversity is guaranteed by the high probability associated with mutation). The same operator is used to determine the solutions to be disregarded from the population at the end of each generation (used inversely: being randomly chosen from the two higher ranks and, the one with the higher rank, or the lower cd, if ranks are the same, being selected to leave the population).

The remainder of the algorithm follows the usual genetic algorithm structure (see Figure 4.2) and other main components are detailed as follows; with all parameters presented hereafter being tuned experimentally.

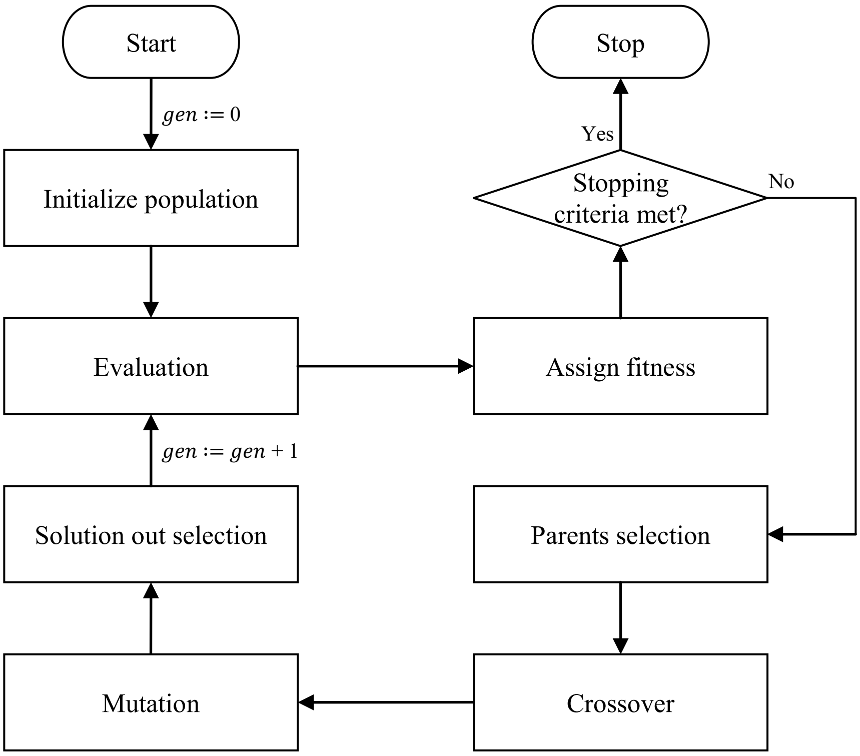


Figure 4.2 Flowchart of the working principle of a genetic algorithm.

Chromosome Representation

The chromosome in the proposed NSGA-II represents a complete solution, being the collection of routes. Both the route (gene) length and the chromosome length are variable and depend on the number of clients serviced and the number of routes in the solution. For example, given a CLRP with 15 clients $I = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$ and 4 possible depot locations $J = \{16, 17, 18, 19\}$, the chromosome representation of a feasible solution is provided as follows.

Solution 1:

16 1 3 4 7 10 16

18 2 6 18

18 5 9 8 13 18

16 11 12 15 14 16.

This solution represents installing facilities 16 and 18, and servicing the clients in the given order by four routes (vehicles).

The adopted representation allows to later build feasible children solutions, thus avoiding the need to use repair methods to restore feasibility (which would increase the processing time and complicate the algorithm).

Initial Population

Following most of the published genetic algorithms, the initial population pop is made of random chromosomes, but including good solutions to accelerate convergence. To obtain the set of good solutions, the randomized extended savings heuristic presented by Prins et al. (2006) for the single-objective CLRP is used. The initial population is composed of 100 solutions ($|pop| = 100$) from which half are random chromosomes and the remaining half obtained using the randomized extended savings heuristic.

Crossover Operator

The crossover operator tries to copy complete routes from the parent to the child, thus will be named route copy crossover (RCX). It operates by copying to the child a random number of routes (between 1/3 and 2/3) from one of the parents, and the remaining unvisited clients are placed in a relocation pool following the original order in the other parent. The clients in the relocation pool are then inserted in the child, in new routes, and using the currently open depots (as long as capacity is obeyed, randomly opening a new one otherwise). This prevents the use of repair methods as child solutions are always feasible. An illustration of the RCX is provided as follows.

Parent 1:	**Parent 2:**
16 1 3 4 7 10 16	17 3 <u>2</u> 4 7 17
18 2 6 18	19 1 <u>6</u> 13 19
18 5 9 8 13 18	18 9 <u>12</u> 8 18
16 11 12 15 14 16.	18 <u>11</u> 5 <u>14</u> 10 <u>15</u> 18.

Assuming the first and third routes of Parent 1 are selected, both are copied to Child 1.

Child 1 (partial encoding):

16 1 3 4 7 10 16

18 5 9 8 13 18.

The remaining clients not yet included in Child 1 (shown underlined in Parent 2) are copied, following their order of appearance, to the relocation pool.

Relocation pool:

2 6 12 11 14 15.

The clients in the relocation pool are then used to form new routes in Child 1, using the currently open depots (opening more when depot capacity constraints are violated) and following the sequence as long as vehicle (route) capacity is obeyed.

Child 1:

16 1 3 4 7 10 16

18 5 9 8 13 18

16 2 6 12 16

16 11 14 15 16.

The second child is created similarly, but using the parents in reverse roles. The RCX thus allows to inherit some of the routes from the parent while at the same time randomizing the building of the child solution routes (yet still partially inheriting the structure of the route from the other parent). Moreover, the operator promotes solutions with few open depots and routes with low available capacity, two features often found in good solutions (cost wise).

Mutation Operators

As mentioned previously, the RCX is more inclined to obtaining low cost solutions, both regarding the number of depots to install as well as regarding the tracing of the routes. As the remaining objectives do not depend on the routes structure (but rather on the installed depots), the mutation operators are more focused on changing the depot structure of the solutions.

Two mutation operators were developed, one directed at changing the routes, and the other more oriented at changing the depots to be installed.

The first operates by randomly swapping the position in the tracing of the routes of two clients (possibly between two routes, depot and vehicle capacity allowing), hence named swap mutation operator. This operator aims at providing a small randomization to the building of the routes. In the following example the operator is applied on the underlined clients.

Solution 1:	**Solution 1 (after swap mutation):**
16 1 3 4 7 10 16	16 1 3 4 9 10 16
18 2 6 18	18 2 6 18
18 5 9 8 13 18	18 5 7 8 13 18
16 11 12 15 14 16.	16 11 12 15 14 16.

The second, named add mutation operator seeks to avoid a fast convergence to solutions with few depots (prone to happen due to the RCX operator), at the same time that diversifies the open depots, by opening a new one and reassigning some routes to it, depot capacity allowing (see following example with changes underlined).

Solution 1:	**Solution 1 (after add mutation):**
16 1 3 4 7 10 16	16 1 3 4 7 10 16
18 2 6 18	18 2 6 18
18 5 9 8 13 18	19 5 9 8 13 19
16 11 12 15 14 16.	19 11 12 15 14 19.

The mutation operators are applied to all the population with a percentage of, respectively, 5% and 50% for the first and second operators. These percentages aim at obtaining a large number of non-dominated solutions, in which the high probability of the second operator allowed to, in fewer generations, significantly diversify the open depots.

Stopping Criterion

In the proposed algorithm the stopping criterion is based on the number of generations gen, where after a fixed number is attained, the algorithm stops. Unlike most implementations of the NSGA-II (where usually only the final set of efficient solutions is returned) non-dominated solutions are obtained and stored/updated at the end of each generation. Although this represents an increased computational burden, it prevents the loss of good solutions during the course of the algorithm as: the initial population often already possesses good solutions (regarding cost); there is a high mutation rate (applied to all the population); and the solution quality of the population is allowed to deteriorate to potentiate the search for the full Pareto front.

After 1000 generations ($gen = 1000$), the algorithm is stopped and the set of stored non-dominated solutions is returned.

4.3.3 Computational Results

An implementation of the proposed NSGA-II for the defined multi-objective CLRP is tested and discussed in the following subsections, where: implementation issues, used test instances, and quality metrics are described; results are presented and discussed; and a graphical example is provided.

Implementation, Tested Instances, and Evaluation

The multi-objective metaheuristic presented here was implemented in C# and results were obtained using a 3.00 GHz Intel Xeon E5450 Quad Core CPU with 8 GB of RAM and Windows XP (without parallel processing).

For the single-objective CLRP benchmark instances are available in the literature (explanation on the instances can be found in Section 3.1.4), still this is not the case for the multi-objective counterpart. Thus, in order to analyse results, some instances were drawn from the single-objective literature, namely from the benchmark set by Barreto et al. (2007). This set was selected as it is based on real-world instances, unlike the remaining sets which were randomly generated. From the chosen set, some instances were disregarded due to having few possible depot locations (instances Perl83-12x2, Perl83-318x4a, and Perl83-318x4b) and thus not prone to exist a reasonable number of non-dominated solutions. Moreover, it is considered that the size s_i of each community (client) $i \in I$ is equal to its demand (d_i). It should be noted that, although the problem structure can fit the multi-objective approach, the original instances were directed at obtaining the location of desirable facilities. Here, the depots to locate are undesirable, so these constitute mere test examples as, for the same problem structure (regarding clients) the depot locations to be considered could/should be different.

Regarding evaluation of methods, whereas on single-objective optimization usually only the value of the objective function (and possibly CPU time) are relevant to access the efficiency of a given algorithm; on multi-objective optimization several other quality metrics have been used, either to evaluate the quality of the front 1 at the end of a given algorithm, or to provide a relative comparison between the final fronts obtained by two algorithms. Moreover, the main inherent evaluation ideas are (Jaszkiewicz, 2004): the ability to provide a good approximation of the exact non-dominated set, and the ability to obtain diversity in the sets of non-dominated solutions (as it is not known the DM's preferences). Thus, the proposed NSGA-II method will be analysed according to one of the objectives (cost, as lower and upper bounds already exist in the literature), and to the number of non-dominated solutions obtained.

Results

In order to obtain results, twenty runs were performed on each instance, from which was obtained: the best solution, cost wise; the number of overall non-dominated solutions; and the average computing time for each instance.

The results can be seen in Table 4.1. The first column displays the name of the instances, followed by the lower bound for the first objective function (LB Z^1). Then, data regarding the proposed NSGA-II are shown, namely, the best obtained cost value (Best Z^1), the number of non-dominated solutions regarding all the runs of the algorithm (NDS) and the average CPU time in seconds. Finally, Gap $_{\text{LB } Z^1}$ refers to the gap, in percentage, between the lower bound and the best obtained result for the first objective function, and Gap $_{\text{Best AGS}}$ concerns the gap (also in percentage) between the best obtained result and the best result from the active guided search (AGS) metaheuristic (Table 3.2). When a value is found with an asterisk it means it is the optimal cost (first objective) of that specific instance. Average and median values are also provided as data showed, for CPU time, Gap $_{\text{LB } Z^1}$, and Gap $_{\text{Best AGS}}$, skewed distributions and/or outlying data points.

Table 4.1 Results of the proposed NSGA-II for some of the instances by Barreto et al. (2007).

	Instance	LB Z^1	NSGA-II Best Z^1	NDS	CPU	Gap $_{LB\ Z^1}$	Gap $_{Best\ AGS}$
1	Christofides69-50x5	551.1	615.0	9	4.7	11.59	8.70
2	Christofides69-75x10	791.4	881.7	19	6.6	11.41	4.23
3	Christofides69-100x10	818.1	894.1	14	10.2	9.29	5.95
4	Daskin95-88x8	347.0	387.7	13	8.6	11.73	5.12
5	Daskin95-150x10	43406.0	45883.4	18	32.6	5.71	3.09
6	Gaskell67-21x5	*424.9	*424.9	4	3.6	0.00	0.00
7	Gaskell67-22x5	*585.1	*585.1	3	4.3	0.00	0.00
8	Gaskell67-29x5	*512.1	517.9	7	4.2	1.13	0.54
9	Gaskell67-32x5a	*562.2	562.8	7	4.1	0.11	0.11
10	Gaskell67-32x5b	*504.3	504.6	6	5.0	0.06	0.06
11	Gaskell67-36x5	*460.4	470.8	5	4.3	2.26	2.26
12	Min92-27x5	*3062.0	3065.2	4	4.1	0.10	0.10
13	Min92-134x8	5423.0	6237.8	18	18.7	15.02	8.88
14	Or76-117x14	12048.4	13254.1	24	14.9	10.01	6.70
16	Perl83-55x15	1074.8	1159.8	4	5.5	7.91	4.27
17	Perl83-85x7	1568.1	1704.5	5	8.6	8.70	4.88
				Average	8.8	5.94	3.43
				Median	5.3	6.81	3.70

Average and median computing times are, respectively, of 8.8 seconds and 5.3 seconds, comparing favourably with other algorithms. A direct comparison with the CPU times of the AGS metaheuristic (corresponding data can be found in Table 3.2) is given in Figure 4.3, showing the NSGA-II to obtain overall faster results.

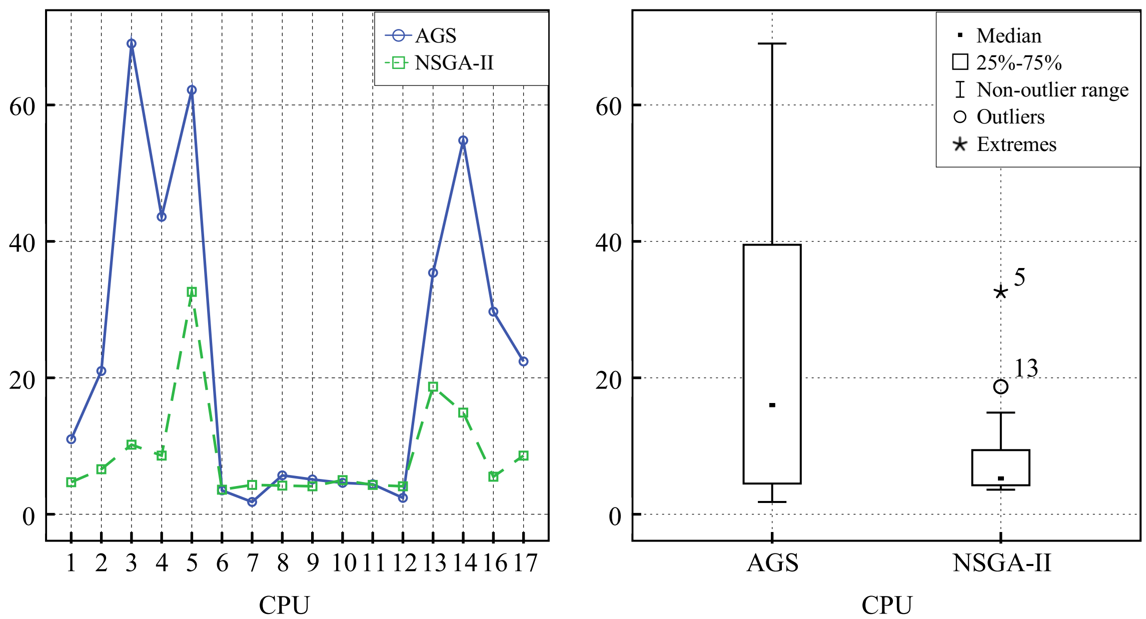


Figure 4.3 Line plots (left) and boxplots (right) for the CPU time, in seconds, concerning the AGS and NSGA-II metaheuristics, Tables 3.2 and 4.1.

Looking at the results, the obtained best solutions regarding cost are, in average, around 6% to the lower bounds and 3.4% to the best AGS results, while median values are, respectively, of 6.8% and 3.7%. Furthermore, two of the known optimal solutions were attained by the proposed NSGA-II. Figure 4.4 depicts a comparison, for the NSGA-II metaheuristic (concerning the cost objective, Z^1), between the gap to the lower bounds and best AGS results (left), and between the gaps to lower bounds of both metaheuristics (right). Figure 4.4 suggests low-quality cost lower bounds for some instances, as the gaps to lower bound of both metaheuristics present similar behaviour.

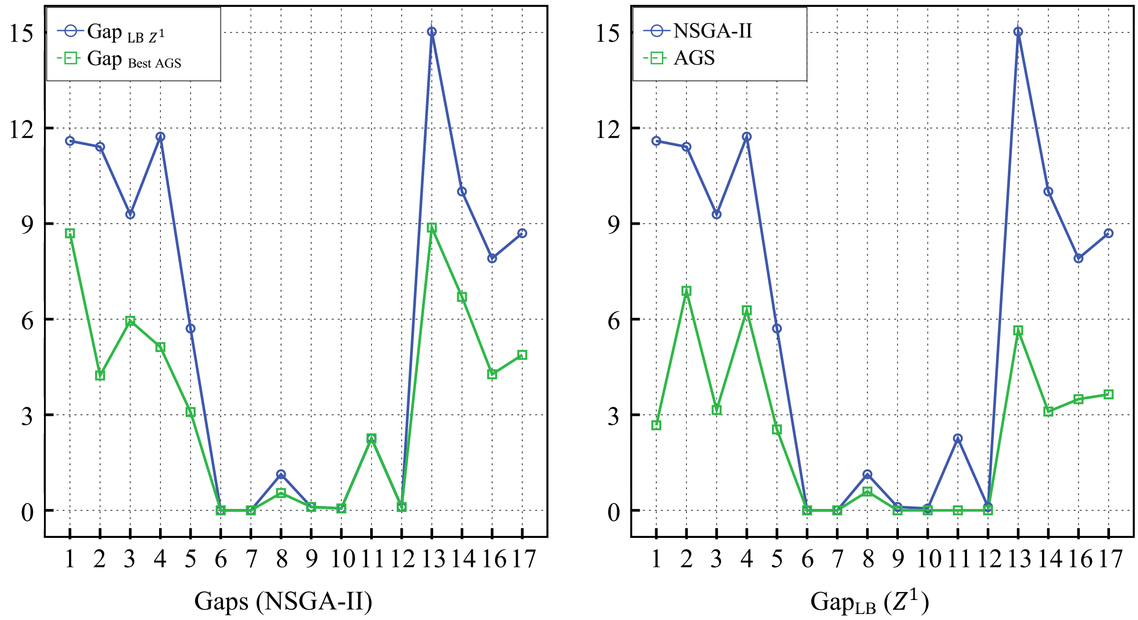


Figure 4.4 Line plots for the gaps, in percentage, between the results from NSGA-II regarding the objective function Z^1 and the best AGS results (left), and for the Gap$_{LB}$ (of Z^1), in percentage, concerning the AGS and NSGA-II metaheuristics (right), Tables 3.2 and 4.1.

Finally, looking at Table 4.1, it can also be concluded that the number of non-dominated solutions increases with the number of possible depot locations, thus, when addressing a real-world scenario, a previous narrowing of the locations to consider may prove fruitful (as it allows to reduce both the computation and cognitive burden).

Graphical Example

In order to further understand specific real-world scenarios, the use of graphical examples may prove adequate and useful. As follows, in order to better understand the three objective model, an instance is selected, from which non-dominated solutions are obtained and results analysed. The instance chosen to be studied is the Gaskell67-21x5 instance from the previously tested set.

For this instance, the previously presented NSGA-II obtained 4 non-dominated solutions, which can be seen in Table 4.2 where, for each solution, objective function values are displayed. When an asterisk is found it indicates optimality regarding the objective function; when underlined, it points to the best overall result in the corresponding objective.

Table 4.2 Obtained non-dominated solutions for instance Gaskell67-21x5.

Solution	Z^1	Z^2	Z^3
S^1	*$\underline{424.9}$	139.81	0.20049
S^2	443.4	130.78	0.20051
S^3	461.6	139.55	$\underline{0.02110}$
S^4	480.4	$\underline{130.52}$	0.02232

Looking at the obtained solutions, several conclusions may be drawn.

Firstly, although cost (Z^1) increases from S^1 to S^4, the increase is of at most 13%; similarly, the overall obnoxious effect (Z^2) has a variation of 7%; however, regarding equity (Z^3), where the maximum individual obnoxious effect is minimized, solutions S^3 and S^4 are only 10% of the other solutions. Hence, if the DM seeks a more equitable solution, it does not incur in a significant increase in both cost and obnoxious effect.

Secondly, obnoxious effect and equity seem positively correlated (as Melachrinoudis et al., 1995, suggest), while cost is negatively correlated with the remaining two.

Also, looking at the solutions in the objectives space (Figure 4.5), regarding equity, two different types of solutions are found (with low and high equity values), each with two different solutions, mostly differing between them the overall obnoxious effect.

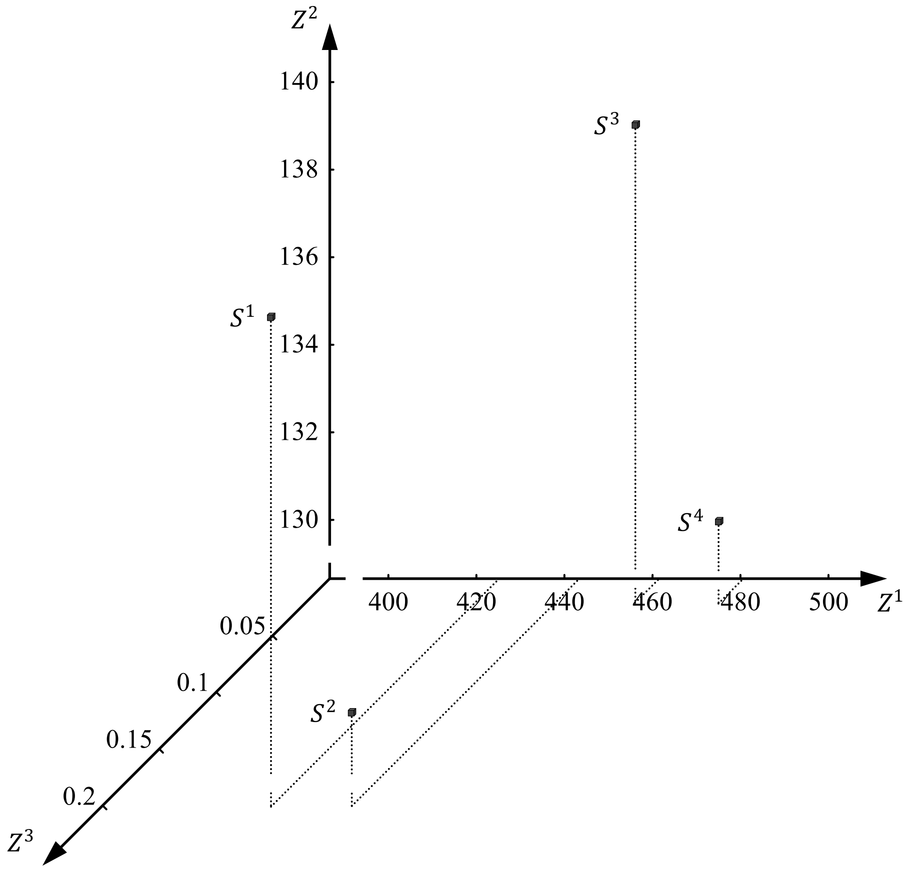


Figure 4.5 Obtained non-dominated solutions in the objectives space (instance Gaskell67-21x5).

Finally, the graphical representation of the efficient solutions (Figure 4.6) shows that, although the routes of S^1 and S^3 have the same tracing (only the depots to open are different, and consequently, their links to clients), this does not hold for S^2 and S^4. This validates the use of an integrated location-routing multi-objective approach, as the routes (and corresponding cost) may change significantly when different depots are considered to be installed.

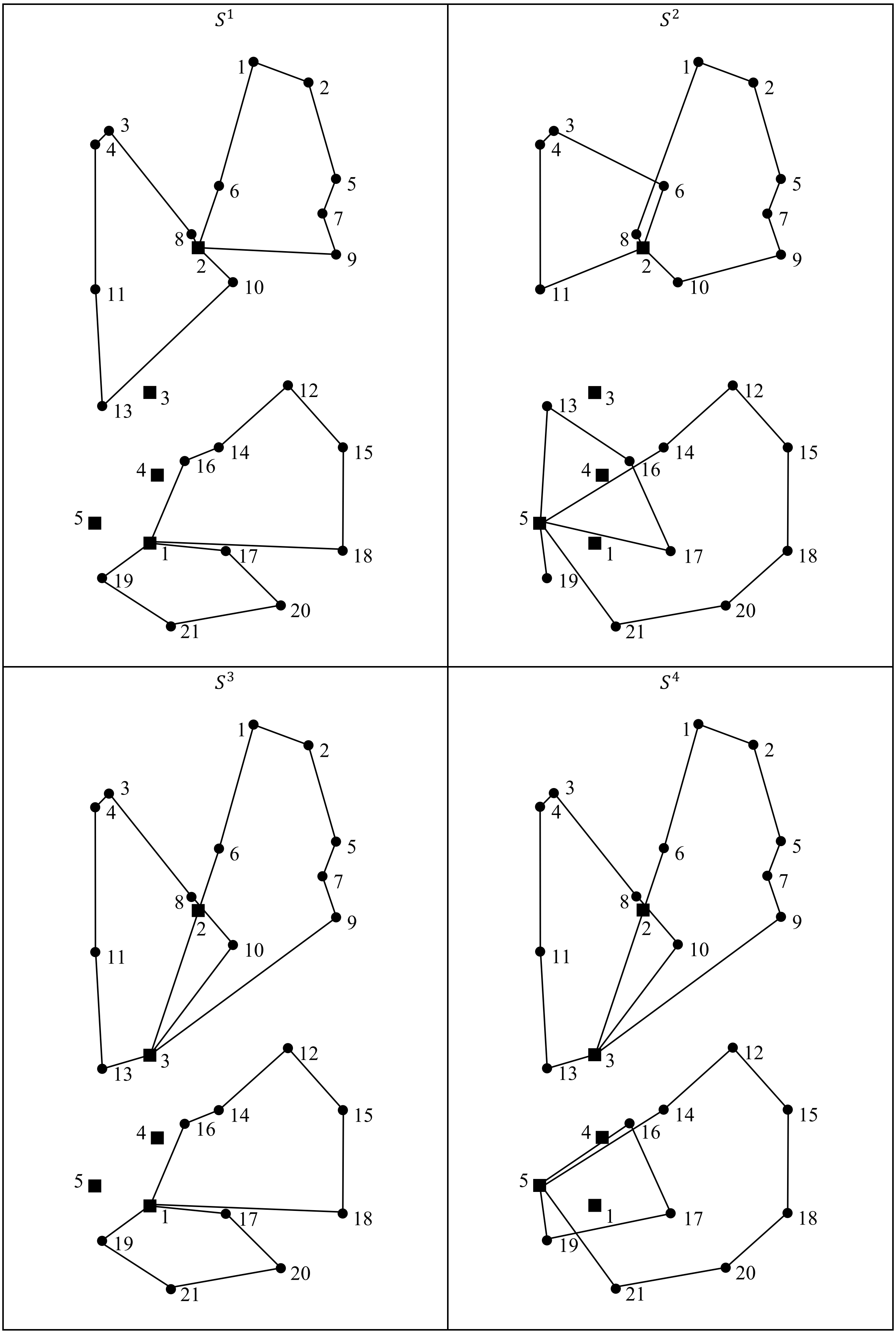


Figure 4.6 Graphical representation of the obtained efficient solutions (instance Gaskell67-21x5).

4.4 Summary

In this chapter, the location-routing of semi-obnoxious facilities was mainly addressed. Firstly, a review on the location of obnoxious and semi-obnoxious facilities was made, where problems were separated according to the number of objectives (one or several), and the number of facilities to install (single or multiple facilities).

Then, works handling both location (of undesirable facilities) and routing were reviewed. When routing is considered, the transported product can be hazardous or not, and, when models handle hazardous products, routing is also undesirable (thus, passing through population centres and long trips should be avoided). Regarding routing of non-hazardous products (and undesirable facility location), as no work exists in the literature, a formal (multi-objective) definition is given based on the CLRP.

The defined multi-objective CLRP is then solved using an evolutionary algorithm. Results are presented and discussed. Furthermore, the non-dominated solutions of a specific instance are analysed, and conclusions drawn regarding the usefulness of multi-objective approaches for the location-routing of this type of facilities.

Chapter 5

Decision Support in Multi-Objective Mixed Integer Programming

A multi-objective mixed integer programming (MOMIP) problem is a mathematical programming problem which considers more than one objective function and some variables are constrained to be integer (being either binary or taking on general integer values). If all variables are integer it is a multi-objective integer programming (MOIP) problem.

As seen previously (see Chapter 4), the location-routing of semi-obnoxious facilities is inherently multi-objective, thus it should be modelled using MOMIP/MOIP. In Chapter 4 a metaheuristic is presented in order to attempt to obtain the whole set of non-dominated solutions. This, in some cases, may not be the best approach as the number of solutions may be too high and, some of them, uninteresting to the decision maker (DM). Here, the decision support for MOMIP will be addressed with different approaches being discussed, and a new method proposed and applied to a small example of a multi-objective capacitated location-routing problem (CLRP).

5.1 Introduction to Multi-Objective Mixed Integer Programming

Mixed integer programming (and the special case, "pure" integer programming) has been used recurrently in many applications. Often, models incorporate discrete values, requiring the consideration of integer variables. Examples include modelling: investment choices, production levels, logical conditions, and location analysis.

The use of these variables brings additional difficulty to models (when comparing with linear programming), as the feasible set is no longer convex. The complexity further increases when multi-objective is considered, as non-dominated solutions exist in the duality gaps. Unlike single-objective approaches, where the optimal solution can be obtained by ordering all the feasible solutions based on the objective function value, in multi-objective, the concept of optimal solution is substituted by the notion of Pareto optimality. In these cases, DMs rather than looking for the optimal solution, they are looking for the "most preferred" solution: the one which (s)he is convinced is the best one.

Therefore, the complete ordering of feasible solutions assumes knowledge about the DM's preferences (the focus of this work being on scenarios where the DM is a single person or a group of people sharing the same objectives and preferences).

Based on this, multi-objective methods can be classified according to the DM's degree of intervention in the solution-finding process (Shin and Ravindran, 1991):

- the DM has full knowledge regarding her/his preference function (*a priori* articulation of preferences)
- the DM does not possess any information regarding the preference function (*a posterior* articulation of preferences)
- partial information is obtained progressively from the DM (progressive articulation of preferences).

The three cases correspond to the different types of information flow found in the decision-making process. In the first case, as the DM states her/his preferences *a priori*, the information flow is from the DM to the method (even though an intermediate person may exist in the process: the analyst). In the *a posteriori* case, the information flow is from the method to the DM, where the preference is stated in the DM's final decision. Finally, in the progressive articulation of preferences, information flows in both directions, from the method to the DM and vice versa, usually with a progressive reduction of the number of choices until the final solution (and corresponding decision) is obtained.

The first two cases encompass methods named non-interactive, as no interaction exists between the DM and the used method. For this reason, methods aim at obtaining the whole set (or a subset) of the non-dominated solutions (e.g. the NSGA-II presented in Chapter 4). As they are designed to generate the whole set or a subset of non-dominated solutions, they may require increased computation time. Moreover, often a large (or even overwhelming) number of solutions is obtained and presented to the DM, adding difficulty to the task of analysing solutions and choosing one.

In the latter case, as methods are characterized by the alternation between human intervention and computation phases, they are named interactive methods. These methods enable reducing the computational effort, as the set of non-dominated solutions is obtained by a progressive articulation of the DM's preferences (thus, only a part of them needs to be generated and evaluated). Preference elicitation (human intervention/decision phase) and solution generation (computation/optimization phase) alternate until the DM considers to have sufficient knowledge of the non-dominated set. In each iteration, the DM is provided with some information and is asked to evaluate the proposed solutions, or to provide additional information regarding her/his preferences (Miettinen et al., 2008).

Figure 5.1 depicts a general framework for an interactive method, where it can be seen that single-objective models are part of multi-objective models.

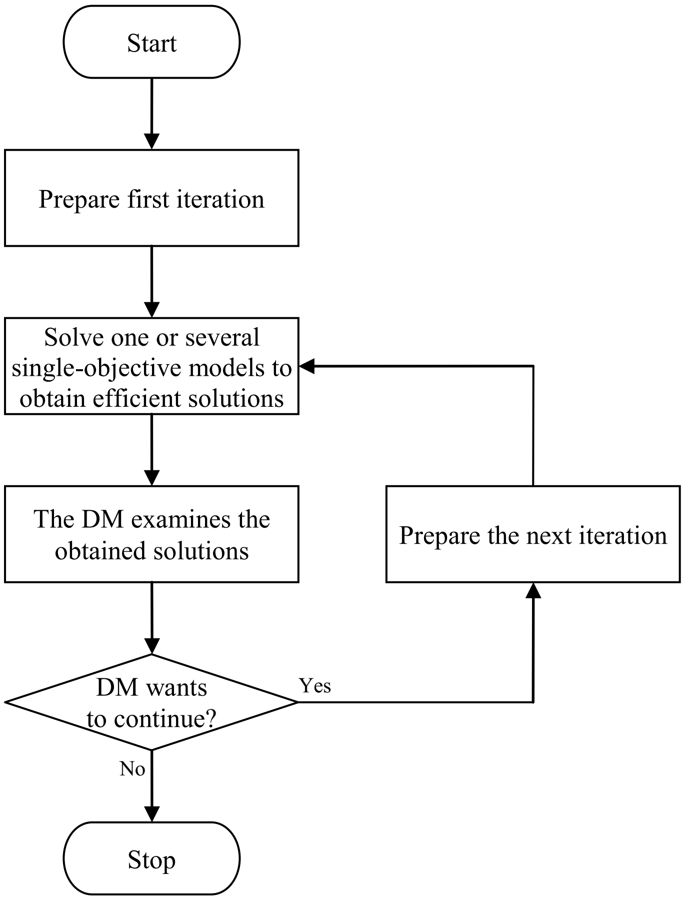


Figure 5.1 Flowchart of the general framework of an interactive method.

When comparing interactive with non-interactive methods, the main difference lies in the involvement of the DM in the solution-finding process. Not only does interactive methods allow the DM to specify and correct her/his preferences during the process, but also it is not required to exist a prespecified global preference structure. This constitutes an important benefit of these methods, as the DM can learn during the process, increasing the knowledge about the problem, its trade-offs, possibilities and limitations (which is often valued by DMs).

Thus, interactive methods overcome weaknesses of *a priori* and *a posteriori* (non-interactive) methods by not requiring a global preference structure from the DM, and generating only non-dominated solutions found to be interesting to the DM (reducing computational effort as well as avoiding the need to compare several non-dominated solutions simultaneously). Moreover, the underlying idea of interactive methods constitutes the major motivation of contemporary decision support systems (Sayin, 2009). Still, some of the advantages can also be seen as drawbacks: as interactive methods rely greatly on the information provided by DMs, and may lack the global view of the non-dominated set, DMs with less knowledge of the situation at hand may end up choosing a less preferred solution.

Overall, each of methods may be more or less appropriate according to the specific decision-making scenario. While a non-interactive method was proposed in the previous chapter, here the focus will be on interactive methods.

As follows, some basic formulations and concepts regarding multi-objective programming will be presented providing background for some of the issues raised in subsequent sections.

5.1.1 Multi-Objective Problem: Formulations and Concepts

Consider the following multi-objective problem:

$$
\begin{array}{lll}
\text{(MOP)} & \max & Z^1 = f^1(x) \\
 & \dots & \\
 & \max & Z^k = f^k(x) \\
 & \text{s.t.:} & x \in X.
\end{array}
$$

$X \subset \mathbb{R}^n$ denotes the non-convex set of feasible solutions defined by the set of functional constraints, $x \geq 0$ and x_j integer for $j \in J \subseteq \{1, 2, ..., n\}$. X is assumed compact (closed and bounded) and non-empty. If all variables of MOP are integer, the problem is considered a multi-objective integer problem; multi-objective mixed integer problem otherwise.

As is assumed the existence of k objective functions, the complete ordering of feasible solutions is not possible, emerging the concepts of efficiency and non-dominance (following Alves and Clímaco, 2009).

A solution $\bar{x} \in X$ is efficient for the MOP if and only if there is no $x \in X$ such that $f^i(x) \geq f^i(\bar{x})$ for all $i \in \{1, 2, ..., k\}$ and $f^i(x) > f^i(\bar{x})$ for at least one i.

A solution $\bar{x} \in X$ is weakly efficient for the MOP if and only if there is no $x \in X$ such that $f^i(x) > f^i(\bar{x})$ for all $i \in \{1, 2, ..., k\}$.

Let $Z \subset \mathbb{R}^k$ be the image of the feasible region X in the objective functions (criteria) space. A point $\bar{z} \in Z$ is called (weakly) non-dominated if it corresponds to a (weakly) efficient solution $\bar{x} \in X$. The terms "efficient", "non-dominated" and "Pareto optimal" are often used indistinctively as synonymous.

A non-dominated point (solution) $\bar{z} \in Z$ is called unsupported if is dominated by a convex combination (not belonging to Z) of other non-dominated criterion points (belonging to Z). As the feasible region is non-convex, unsupported non-dominated solutions may exist (lying inside the duality gaps). Thus, unlike in multi-objective linear programming, the set of non-dominated solutions of MOP cannot be fully obtained by varying the parameter λ on the weighted sum of the objective functions:

$$
\begin{array}{lll}
(\text{MOP}_\lambda) & \max & \sum_{i=1}^{k} \lambda^i f^i(x) \\
 & \text{s.t.:} & x \in X
\end{array}
$$

where $\lambda \in \Lambda = \{\lambda \in \mathbb{R}^k : \lambda^i > 0\ \forall i, \sum_{i=1}^{k} \lambda^i = 1\}$.

Even if the complete parameterization of λ is attempted, unsupported non-dominated solutions cannot be reached. One way to overcome this is by adding additional constraints into MOP $_{\lambda}$, imposing bounds on the objective function values (Soland, 1979):

$$
\begin{aligned}
(\text{MOP}_{\lambda,\alpha}) \quad \max \quad & \sum_{i=1}^{k} \lambda^i f^i(x) \\
\text{s.t.:} \quad & x \in X, \\
& Z^i \geq \alpha^i \qquad\qquad i = 1, 2, ..., k
\end{aligned}
$$

where $\lambda \in \Lambda$ and $\alpha \in \mathbb{R}^k$.

Every solution obtained by MOP $_{\lambda,\alpha}$ is non-dominated and there is always an α such that MOP $_{\lambda,\alpha}$ returns a specific non-dominated solution. Thus, MOP $_{\lambda,\alpha}$ allows determining the complete set of non-dominated solutions of MOP.

The ideal values (Z^{i^*}, $i = 1, 2, ..., k$) are defined by the maximum criterion values over the set of efficient solutions E, and can be obtained by individually optimizing each objective function. The resulting ideal point $Z^* = (Z^{1^*}, Z^{2^*}, ..., Z^{k^*})$ is usually not feasible, otherwise no conflict exists between criteria and the solution is optimal.

The minimum criterion values over the efficient set E are called nadir values ($\breve{Z}^i$, $i = 1, 2, ..., k$). Unlike the ideal point, the nadir point $\breve{Z} = (\breve{Z}^1, \breve{Z}^2, ..., \breve{Z}^k)$ is generally very hard to obtain (for $k \geq 3$).

Both points (ideal and nadir) provide the range of objective values within which all non-dominated solutions are found. This information is valuable as (Alves and Costa, 2009): it allows DMs to size the multi-objective problem, is relevant for the graphical representation of non-dominated points, and can be used to normalize objectives.

Due to the estimation of the true nadir values being an unsolved computing problem (for more than two objectives), the minimum values of the payoff table are often used. The payoff table comprises the non-dominated solutions which optimize each objective function individually (see Table 5.1). In the ith row of the table it can be found the criterion values ($Z^1(x^i)$, ..., Z^{i^*}, ..., $Z^k(x^i)$) for the efficient solution $x^i \in E$ that maximizes the ith objective ($i = 1, 2, ..., k$).

Table 5.1 Payoff table.

Efficient solution	Z^1	Z^2		Z^k
x^1	$Z^{1^*} = Z^1(x^1)$	$Z^2(x^1)$	...	$Z^k(x^1)$
x^2	$Z^1(x^2)$	$Z^{2^*} = Z^2(x^2)$	...	$Z^k(x^2)$
...	...	...	...	...
x^k	$Z^1(x^k)$	$Z^2(x^k)$	...	$Z^{k^*} = Z^k(x^k)$

The main diagonal of the payoff table provides the ideal point Z^*, while the minimum values for each column provides an estimate for the nadir point $\breve{Z}$. This is in most cases an overestimation of the true nadir point that tends to be farthest from it as the size of the problem increases (Isermann and Steuer, 1988). However, in the bi-objective case ($k = 2$), the true nadir point always corresponds to the minimum value for each column of the payoff table.

5.2 Interactive Multi-Objective Mixed Integer Programming Methods

In the last section the two main type of approaches in multi-objective programming were presented and discussed: non-interactive and interactive methods. As follows, the focus will be on interactive methods for multi-objective mixed integer programming (MOMIP) models (thus also applicable to multi-objective integer programming – MOIP – models), where some of the works in the literature are reviewed. This subject has been surveyed by Evans (1984), Rasmussen (1986), Clímaco et al. (1997), and Alves and Clímaco (2007).

According to Alves and Clímaco (2007) different paradigms exist for interactive methods. Some represent the DM's preferences by an implicit utility function (then, methods try to obtain the best result regarding it, usually requiring no contradictions exists in the human/DM intervention phase). Others aim at a progressive and selective learning of the non-dominated set (open communication protocol). In the latter, the method does not intend to converge to any given solution, but rather help the DM in identifying satisfactory solutions. Furthermore, in the open communication paradigm, there are no irrevocable decisions during the solution-finding process, allowing the DM to return to previous iterations.

The characteristics of open communication protocol methods leads to believe that these may be more fit for current decision-making scenarios (view equally shared by Alves and Clímaco, 2007, 2009). This justifies the study of this type of methods, being the main focus of this section.

For multi-objective methods, a natural separation can be made regarding the number of objectives (due to the inherit difference in complexity). Methods for bi-objective models are not directly applicable for models with three or more objectives as complexity increases (e.g. when obtaining the nadir point). However, models for three objectives are easily adaptable to several other objectives. This leads to the separation between methods for bi-objective and multi-objective models.

Looking at the literature on open communication interactive methods, only the method by Ferreira et al. (1996) is restricted to the bi-objective case. It starts by obtaining the pair of non-dominated solutions that individually optimize each of the objective functions (thus gaining knowledge on the ideal and nadir points, and subsequently on the objectives range) from which a region is defined. Then, at each iteration, the DM is required to choose a pair of non-dominated solutions or specify bounds on the objective function values. Based on this information, a subregion is defined, within which a weighted sum program (MOP $_{\lambda,\alpha}$) is solved to optimality. This allows eliminating areas in the objectives space: by unfeasibility (the complete subregion), if no solution is found; or by dominance and unfeasibility (parts of the original region), if a new non-dominated solution is attained. The use of the Tchebycheff metric, instead of the weighted sum program (otherwise the same), was proposed later by Ferreira (1997), allowing the elimination of larger regions by unfeasibility (when a non-dominated solution is attained). This method requires minimal cognitive effort from DMs.

The remaining of the reviewed methods are able to address multi-objective models and will be discussed as follows.

Durso (1992) modified the interactive branch-and-bound method by Marcotte and Soland (1986) in order to account for mixed integer linear programming. The modification was on the approach used for obtaining non-dominated solutions where, rather than a weighted sum program, an augmented weighted Tchebycheff metric is used. For each node in the branch-and-bound tree, the non-dominated solutions which allow determining the ideal point are obtained. At each iteration the DM chooses the preferred ideal point, thus indicating the node to further analyse. For that node the augmented weighted Tchebycheff metric determines the "central" non-dominated solution (using equal weights for all criteria). The DM is then required to choose the preferred non-dominated solution out of all known in the node under analysis (the parent node), originating the creation of new (child) nodes. The child nodes to be created (at most equal to the number of objective functions) have to be within a range (defined by the DM) to the ideal point of the parent node. Each child node inherits the constraints of the parent node plus new lower bounds resulting from the chosen non-dominated solution. This is a progressive reduction of the non-dominated set, performed until the DM is satisfied with the obtained solutions.

Similar concepts were used in the work by L'Hoir and Teghem (1995) with a method called MOMIX, also applicable to mixed integer linear programming. Following the work by Marcotte and Soland (1986) the method uses an interactive branch-and-bound tree. For each node (of the tree), a non-dominated solution is firstly obtained minimizing a weighted Tchebycheff distance to the corresponding ideal point. The method is then composed of two steps: a "depth first" progression in the tree and a "backtracking" procedure. In the depth first progression step (aimed at obtaining a first solution), for each node, the DM chooses the objective function to improve in priority (and eventually prioritizing all the objectives), upon which a new subnode is created. The new subnode has a new lower bound based on the non-dominated solution obtained by the Tchebycheff program in the node. The backtracking step intends to confirm the degree of satisfaction of the DM with regards to the achieved solution (or to find a new preferred solution if needed). This is done by examining other parts of the tree where new subnodes are generated based on the priority of objectives provided by the DM. According to Teghem (2009), generating more than 2 or 3 new subnodes rarely brings any improved solution, thus, the backtracking procedure does not need to explore the whole tree.

Other open communication interactive methods are the ones by Vassilev and Narula (1993), Narula and Vassilev (1994), and Karaivanova et al. (1995) which, although proposed for multi-objective integer linear programming, according to Alves and Clímaco (2007), are also applicable to the mixed integer case. In the three methods, the same type of information is required from the DM, namely, her/his preferences: the aspiration (reference point) and reservation levels for the objective functions.

The method by Vassilev and Narula (1993) starts by obtaining a non-dominated solution. Then, in the human intervention phase, if the DM is not satisfied with solution, (s)he is required to specify a new reference point with better, worse and equal value(s), respectively, for the objective function(s) to improve, allowed to deteriorate and to maintain unaltered. Based on the reference point and the non-dominated solution, a scalarizing program is solved, being the newly obtained non-dominated solution again submitted to the human intervention phase. Later, Narula and Vassilev (1994) propose a modification in order to reduce the computational effort: rather than integer solutions, continuous solutions are obtained. In the human intervention phase, the DM may continue to search for non-dominated continuous solutions or require the computation of the "closest" integer solution.

Karaivanova et al. (1995) presented two methods with similar concepts. In the first, integer non-dominated solutions are computed, while in the second, both continuous and integer solutions are obtained. The first method has an underlying principle similar to that of Vassilev and Narula (1993) where, instead of maximizing the smallest standardized difference to the solution under analysis (regarding the objective functions to improve), the largest standardized difference to the reference point (for the same objective functions) is minimized. The reference point values of the remaining objective functions are used as lower bounds. The second method uses the Pareto race method (Korhonen and Wallenius, 1988) to obtain new solutions along the continuous non-dominated frontier. When the DM considers to have found the preferred solution for the continuous problem, the integer solution closest to it is computed (as in Narula and Vassilev, 1994). The authors compare both methods and conclude that, while the first is more time consuming, the second produces solutions which may be unsatisfactory to DM. Thus, a system that combines both methods is proposed.

More recently, Alves and Clímaco (2000) presented a method that uses branch-and-bound techniques for mixed integer linear programming. At each iteration, the DM must specify a reference point or choose an objective function intended to improve (regarding the previously obtained non-dominated solution). If the DM opts to use the latter, the reference point's component corresponding to the chosen criterion is automatically adjusted (seen as a directional search). Tchebycheff scalarizing programs are then solved successively by branch-and-bound in order to obtain non-dominated solutions. The use of sensitivity analysis enables to profit from previous iterations, as it allows to gain knowledge on the range to the previous reference point within which the structure of the previous branch-and-bound tree remains unchanged. To continue the search in the same direction the range is surpassed slightly, being the previous tree used to proceed to the next computations. To obtain new branching, an attempt is firstly made to simplify the previous tree, before expanding it until a new non-dominated solution is obtained. With this simplification/branching procedure, the time of computation phases can be reduced.

In interactive methods, main concerns are the computational effort and the cognitive burden to DMs. Methods by Durso (1992) and L'Hoir and Teghem (1995) require to solve at each iteration several mixed integer programs, and thus are very demanding computation wise. Vassilev and Narula (1993), Narula and Vassilev (1994), Karaivanova et al. (1995), and Alves and Clímaco (2000) attempt to reduce the computational effort without placing many questions to the DM. Still, the method by Ferreira et al. (1996) (later extended in Ferreira, 1997) seems to be the least demanding from both computational and cognitive point of view. However, the method's application is limited to bi-objective models.

5.3 An Interactive Method for Multi-Objective Mixed Integer Programming

In the previous section open communication interactive methods for MOMIP were reviewed. From the reviewed methods the most attractive, regarding computational and cognitive burden, is the method by Ferreira et al. (1996). Still, the method has a drawback: its applicability is limited to bi-objective problems.

In this section, a proposal is made with the purpose of extending the method by Ferreira et al. (1996) to the three-objective case (and consequently applicable to more than three objectives) (Ferreira et al., 2010). The proposal should therefore maintain the main goal of the method, that is, obtain solutions posing little computational effort and allowing to search the objectives space with the least cognitive effort.

In the following subsections the proposal will be presented and a step-by-step example will illustrate its application to a MOIP model (although also applicable to MOMIP): a multi-objective CLRP.

5.3.1 Proposal

The proposal presented here to tackle MOMIP searches for non-dominated solutions in the objectives space, allowing the DM to obtain any non-dominated solution (both supported and unsupported). There are no irrevocable decisions throughout the process, and the method is not too demanding regarding the information required from the DM. The general framework of the proposed method is depicted in Figure 5.2 as a flowchart.

The following graphical and numerical information are provided and updated at each iteration.

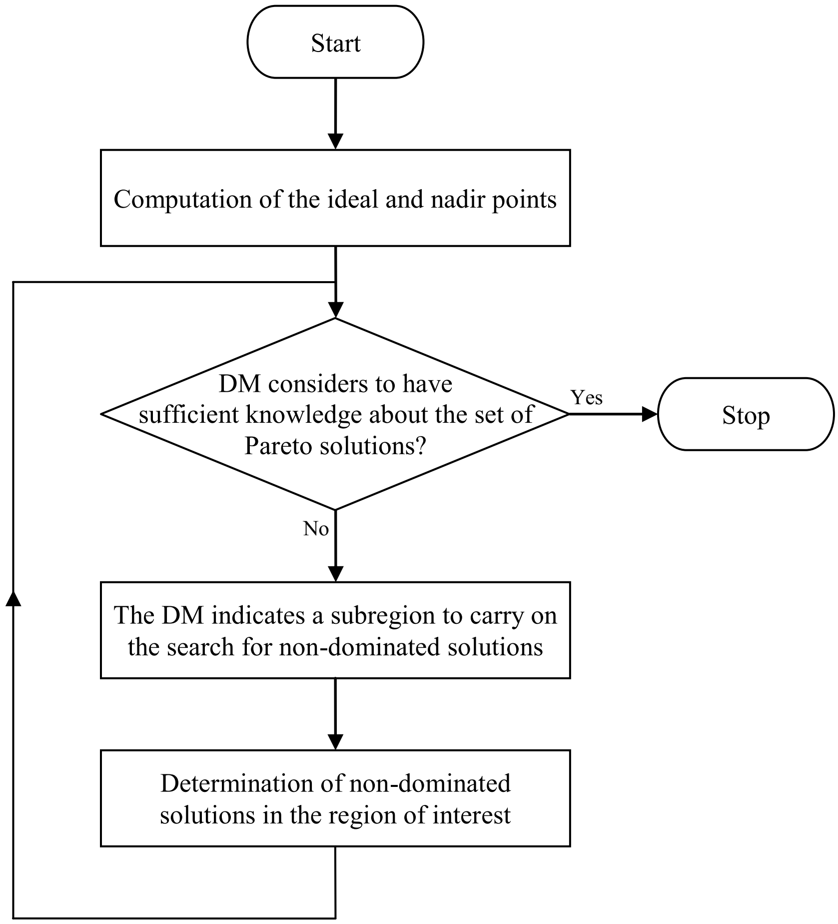
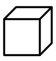
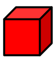


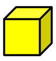
Figure 5.2 Flowchart of the general framework of the proposed interactive method.

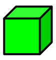
Graphical information in the objectives space:

- the range of values allowed for each objective (obtained with the ideal and nadir points)
- the currently known non-dominated solutions, represented by
- a colour hierarchy regarding the different regions (red overrides yellow which, in turn surpasses green), otherwise explored

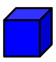
   explored region

   unfeasible/dominated region obtained in the previous iteration

   non-dominated unexplored region (with only one criterion better)

   non-dominated unexplored region (with two criteria better)

- subregion which the DM wants to explore (region of interest)

   region of interest.

Numerical information:

- value of efficient solutions (decision space)
- value of non-dominated solutions (objective).

The method starts out by computing the ideal and nadir points. This is done by individually optimizing each objective function, thus obtaining the payoff table with three solutions (considering $S^l = (Z^1(x^l), Z^2(x^l), Z^3(x^l))$, obtained solutions are S^1, S^2, and S^3), from where the ideal point Z^* and (for simplicity's sake) an estimation of the nadir point $\check{Z}$ are obtained. Using both points the initial region is drawn, representing the (estimated) range of values for each objective function. The resulting region is a rectangular parallelepiped, which is unexplored with regards to the existence of non-dominated solutions, thus assuming the colour green (Figure 5.3).

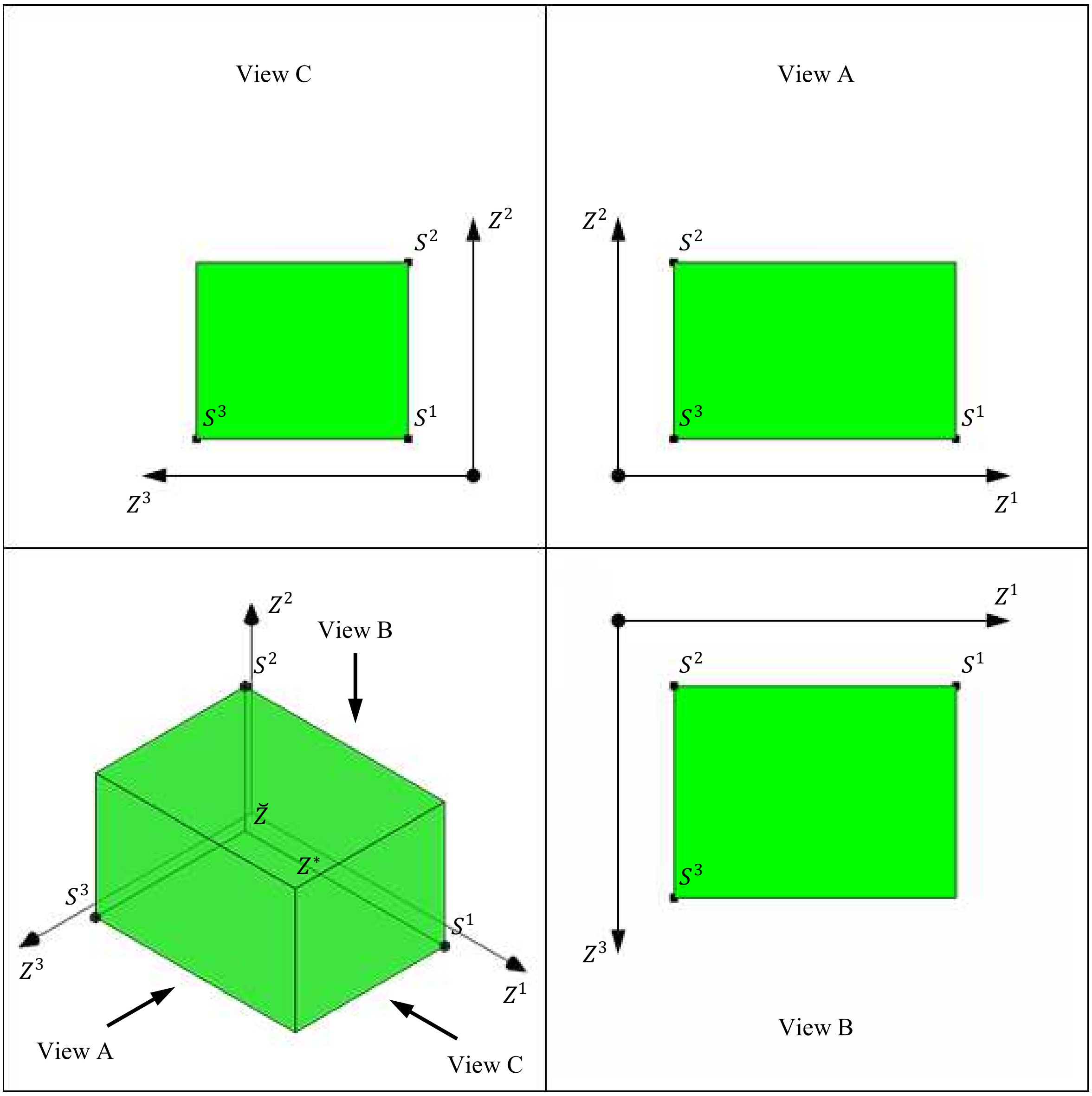


Figure 5.3 Output of the first step of the interactive method.

More often than not, the first three solutions (S^1, S^2, and S^3) allow to draw conclusions with regards to the unexplored region (see Section 5.3.2), as the values for the objective functions, disregarding the ideal and (estimated) nadir values, are usually better than the (estimated) nadir value. For the sake of simplicity, here, it will be assumed that $Z^i(x^l) = \breve{Z}^i$ for $i \neq l$ ($i = 1, 2, 3$ and $l = 1, 2, 3$). When it is not the case, the inclusion of each of the three solutions is handled as an iteration (with the corresponding procedure being described as follows), only lacking the DM intervention (and thus skipping the definition of the region of interest and the computation of the weighted sum program).

Until the DM considers to have sufficient knowledge on the set of non-dominated solutions, the following iterations are performed (depicted in Figures 5.4-5.9).

The DM is required to indicate a subregion to carry on the search for non-dominated solutions. This can be done by: choosing a pair of (currently) adjacent non-dominated solutions or by imposing bounds on the objective functions. In the latter, the bounds can be defined numerically or graphically, having in mind that the definition of the subregion should be done within the non-dominated unexplored region (coloured green or yellow). The chosen subregion is called region of interest and is represented with the colour blue (Figures 5.4, 5.6, and 5.8).

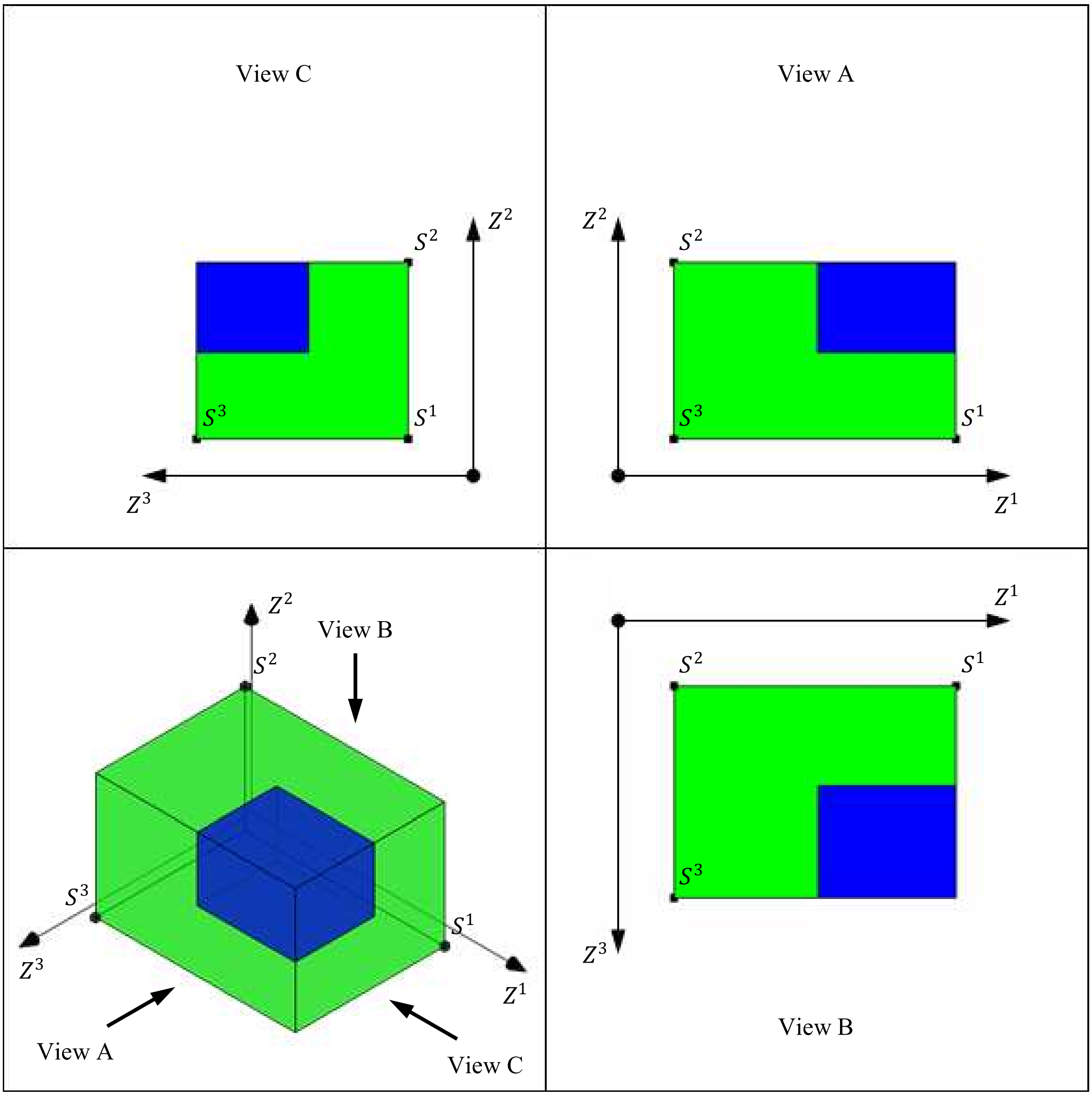


Figure 5.4 First iteration of the method with the DM-defined region of interest.

Within the region of interest a weighted sum program (MOP $_{\lambda,\alpha}$) is solved to optimality with $k = 3$ and the additional constraints:

$$Z^i \geq \alpha^i_{min} \qquad i = 1, 2, 3,$$

$$Z^i \leq \alpha^i_{max} \qquad i = 1, 2, 3.$$

Where α^i_{min} and α^i_{max} are respectively the lower and upper bounds of the (DM-defined) region of interest for the objective function i ($\breve{Z}^i < \alpha^i_{min} < \alpha^i_{max} < Z^{i^*}$, $i = 1, 2, 3$). The resulting non-dominated solution, say S^{k+it} for the itth iteration, allows to characterize and eliminate regions in the objectives space (see Figures 5.5 and 5.7).

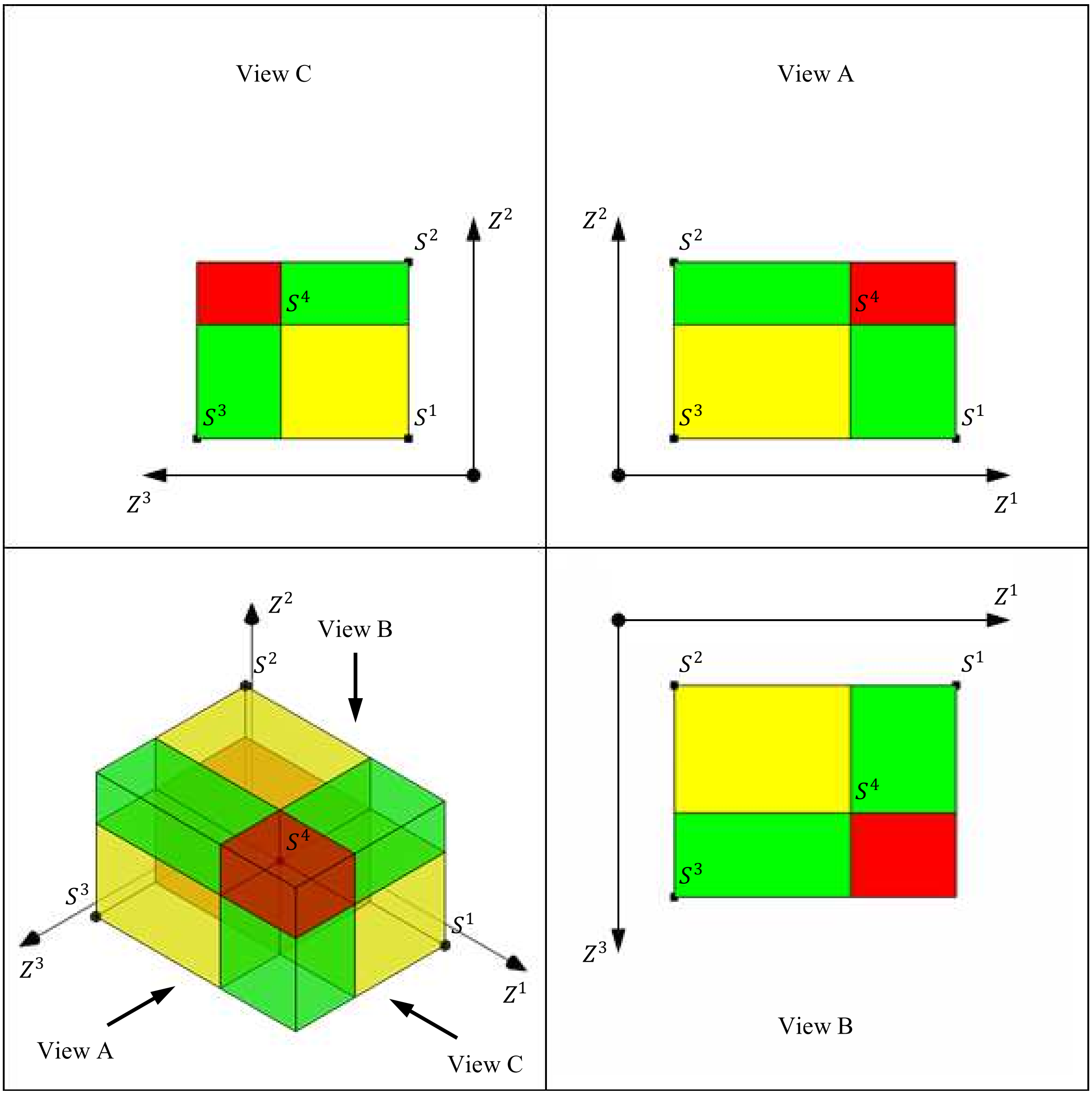


Figure 5.5 First iteration of the method with the region characterization after obtaining S^4.

The first regions to be update are the ones coloured red (found to be unfeasible or dominated in the previous iteration). These are now considered explored and cleared of any colour, in order not to clutter the image. Then, the following non-dominated unexplored regions (currently yellow or green) are coloured red and will be considered explored in the next iteration:

- by unfeasibility, where $Z^1 \geq Z^1(x^{k+it}) \cap Z^2 \geq Z^2(x^{k+it}) \cap Z^3 \geq Z^3(x^{k+it})$
- by dominance, where $Z^1 \leq Z^1(x^{k+it}) \cap Z^2 \leq Z^2(x^{k+it}) \cap Z^3 \leq Z^3(x^{k+it})$.

The non-dominated unexplored regions are separated into yellow and green. Albeit any solution found in these regions using MOP $_{\lambda,\alpha}$ is non-dominated, it was made a separation between regions characterized as: yellow, where found solutions have only one criterion in which is better than any one solution obtained thus far; and green, where found solutions have two criteria in which are better than any currently obtained solution.

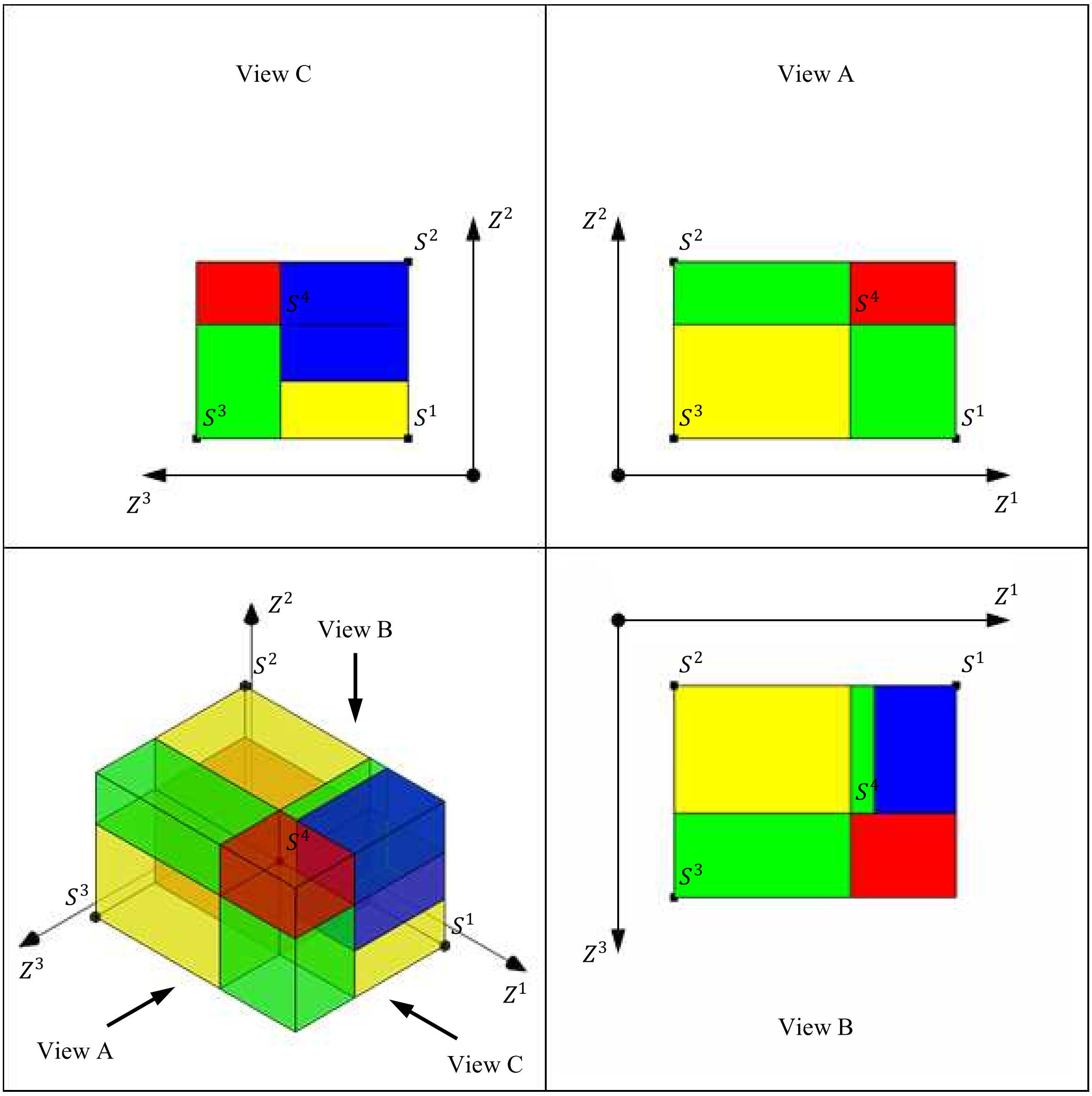


Figure 5.6 Second iteration of the method with the DM-defined region of interest.

By making this separation, the DM is easily aware of regions where found solutions provide a trade-off which (s)he may consider potentially more advantageous.

The regions to be coloured yellow are the currently green ones where:

- $Z^1 \geq Z^1(x^{k+it}) \cap Z^2 \leq Z^2(x^{k+it}) \cap Z^3 \leq Z^3(x^{k+it})$
- $Z^1 \leq Z^1(x^{k+it}) \cap Z^2 \geq Z^2(x^{k+it}) \cap Z^3 \leq Z^3(x^{k+it})$
- $Z^1 \leq Z^1(x^{k+it}) \cap Z^2 \leq Z^2(x^{k+it}) \cap Z^3 \geq Z^3(x^{k+it})$.

The remaining (green) regions stay unaltered. Empirical tests have shown that, often, the green region is quickly reduced, remaining only yellow non-dominated unexplored regions.

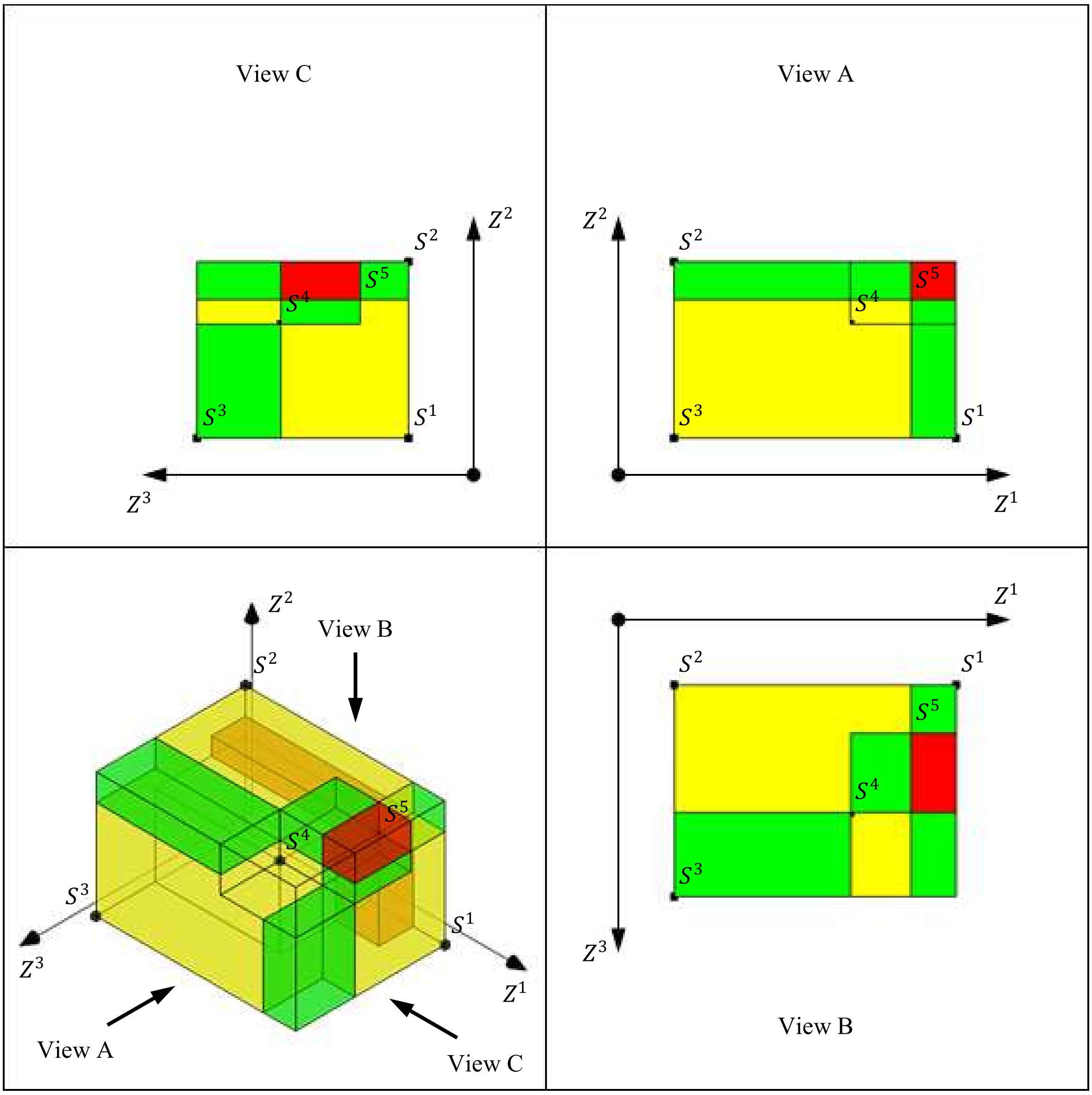


Figure 5.7 Second iteration of the method with the region characterization after obtaining S^5.

The chosen colour hierarchy may enable a fast identification of the different explored/unexplored regions. However, other set of colours may be used for the different regions, providing a hierarchy is defined.

Previously, it was assumed that the weighted sum program (MOP $_{\lambda,\alpha}$) found a (non-dominated) solution in the region of interest. If this is not the case, and no solution was found, conclusions can nevertheless be drawn. In this case, the region of interest is considered explored and cleared of any colour. This procedure can be seen in Figure 5.9 where, in the same iteration, no solution was found in the DM-defined region of interest (subregion seen in Figure 5.8 with colour blue), thus being considered explored.

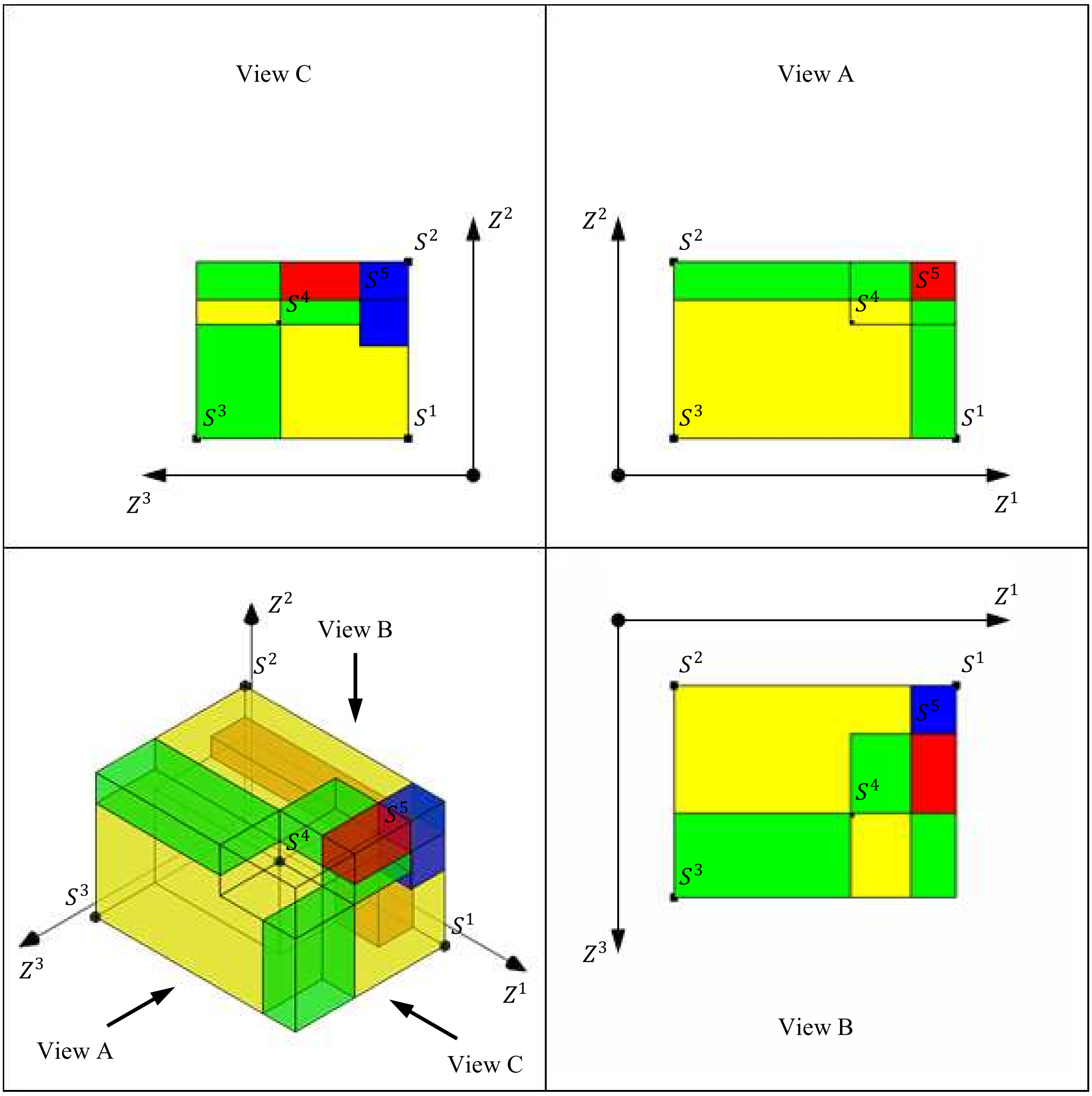


Figure 5.8 Third iteration of the method with the DM-defined region of interest.

The proposed interactive method has several interesting features (inherited from the method by Ferreira et al., 1996), from both the DM as well as the computational point of view.

Regarding the DM choices, none is irrevocable, meaning (s)he can, at all times, return to previous unexplored regions and proceed from there. Also, the information required from the DM in each iteration is not too demanding, as it is only required to be indicated a subregion within which the search for non-dominated solutions is to be continued (called region of interest). The indication of the region of interest can be done by choosing two adjacent non-dominated solutions or by imposing bounds on the objectives (where both graphical and numeric input are allowed).

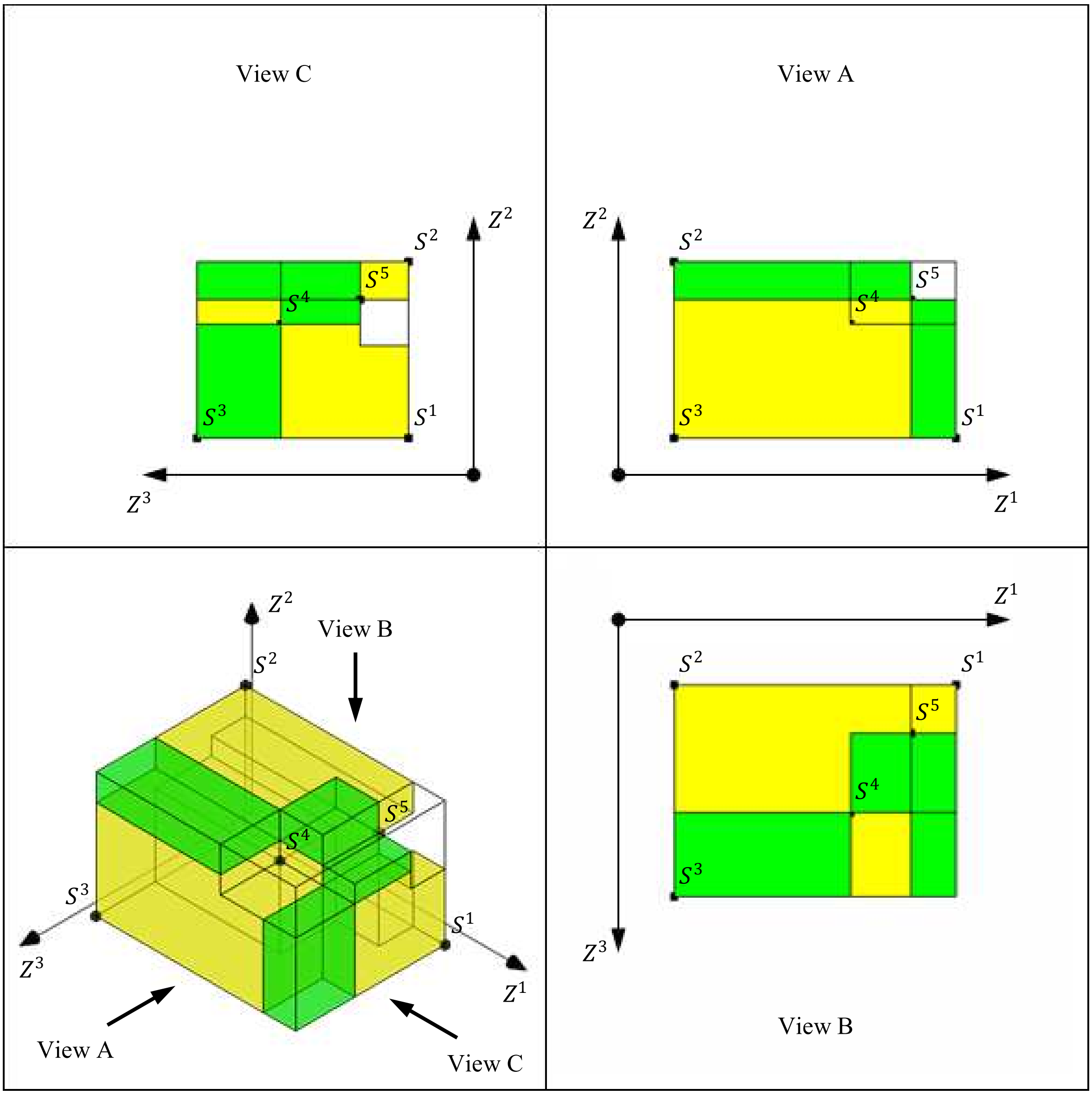


Figure 5.9 Third iteration of the method after no solution was found in the region of interest.

From the computational point of view, a single-objective mixed integer programming problem has to be solved at each iteration. Moreover, as the structure of the problem remains almost unchanged, computational advantages exists.

Finally, the method enables to find any non-dominated solution of the problem (both supported and unsupported).

In order to extend this method to four or more objectives, the graphical representation may become increasingly difficult to interpret. Nevertheless, the procedure remains the same, requiring only to be defined an additional colour in the previously described colour hierarchy (if one wishes to distinguish the different non-dominated unexplored regions). It should be noted, however, that the regions to be eliminated by unfeasibility or dominance are always only two per iteration (when a non-dominated solution is found), which may become less relevant when several objectives are simultaneously considered (the number of different regions obtained in each iteration is 2^k), reducing the usefulness of the method. For example, if five objective functions are considered ($k = 5$), at each iteration 32 different regions are generated, from which only 2 can be eliminated by unfeasibility or dominance.

5.3.2 Step-by-Step Example

In order to further examine the proposed interactive method a step-by-step example is applied to a multi-objective CLRP. The formulation used for the problem follows the CLRP_2 formulation presented in Section 4.3.1, thus all objective functions are to be minimized. Data regarding the test instance can be seen in Appendix B.

The method starts by obtaining the payoff table (as seen in Table 5.2), with three solutions (corresponding graphical representation can be seen in Figure 5.10), from which the ideal point $Z^* = (115.0240,\ 6.8483,\ 0.054717)$ and the (estimated) nadir point $\breve{Z} = (155.5355,\ 18.3037,\ 0.227730)$ are obtained, allowing to define the (estimated) range of values for the objective functions.

Table 5.2 Payoff table of the test instance.

Solution	Z^1	Z^2	Z^3
S^1	115.0240	18.3037	0.227730
S^2	155.5355	6.8483	0.059434
S^3	138.4380	7.9179	0.054717

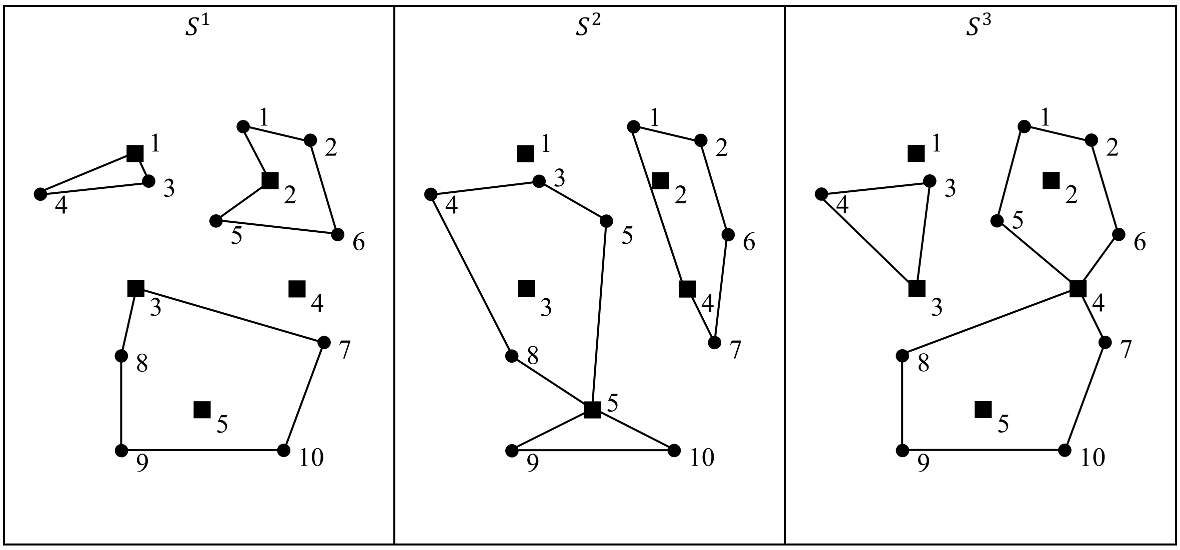


Figure 5.10 Graphical representation of the efficient solutions S^1, S^2, and S^3 of the test instance.

With this information it is possible to draw the initial region (as seen in Figure 5.11). Afterwards, further information can be obtained with the inclusion of the three solutions in the payoff table. Figures 5.12, 5.13, and 5.14 display how the inclusion of, respectively, S^1, S^2, and S^3 allow to categorize the different regions in the objectives space (always having in mind the colour hierarchy, where red surpasses yellow which in turn overlaps green).

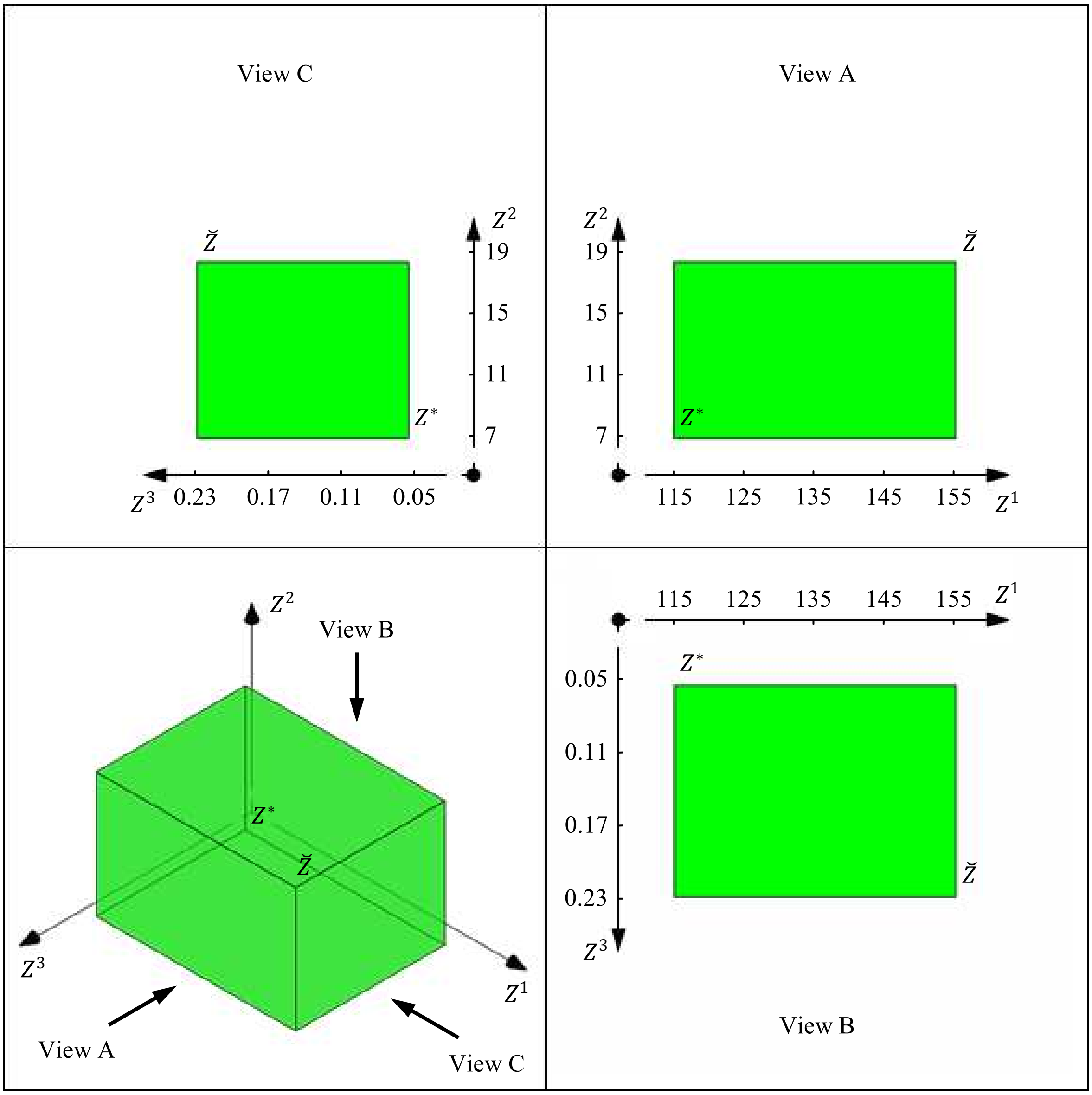


Figure 5.11 Initial region characterization of the method (test instance).

In Figure 5.12, the inclusion of S^1 does not allow to draw conclusions on the remaining regions, as the values for the second and third objective functions correspond to the (estimated) nadir values. Upon including S^2 (depicted in Figure 5.13), as the value for the first objective function corresponds to the (estimated) nadir value, conclusions can only be drawn regarding a region where: any newly found solution is better than any existing solution with regards to only one of the objective functions (Z^1), which happens for $Z^3 \geq 0.059434$, and thus, is coloured yellow. The inclusion of S^3 (Figure 5.14), on the other hand, allows to define several different regions, as the following characterization is made:

- red, by dominance, when $Z^1 \geq 138.4380 \cap Z^2 \geq 7.9179 \cap Z^3 \geq 0.054717$
- yellow, when $Z^1 \leq 138.4380 \cap Z^2 \geq 7.9179 \cap Z^3 \geq 0.054717$
- yellow, when $Z^1 \geq 138.4380 \cap Z^2 \leq 7.9179 \cap Z^3 \geq 0.054717$.

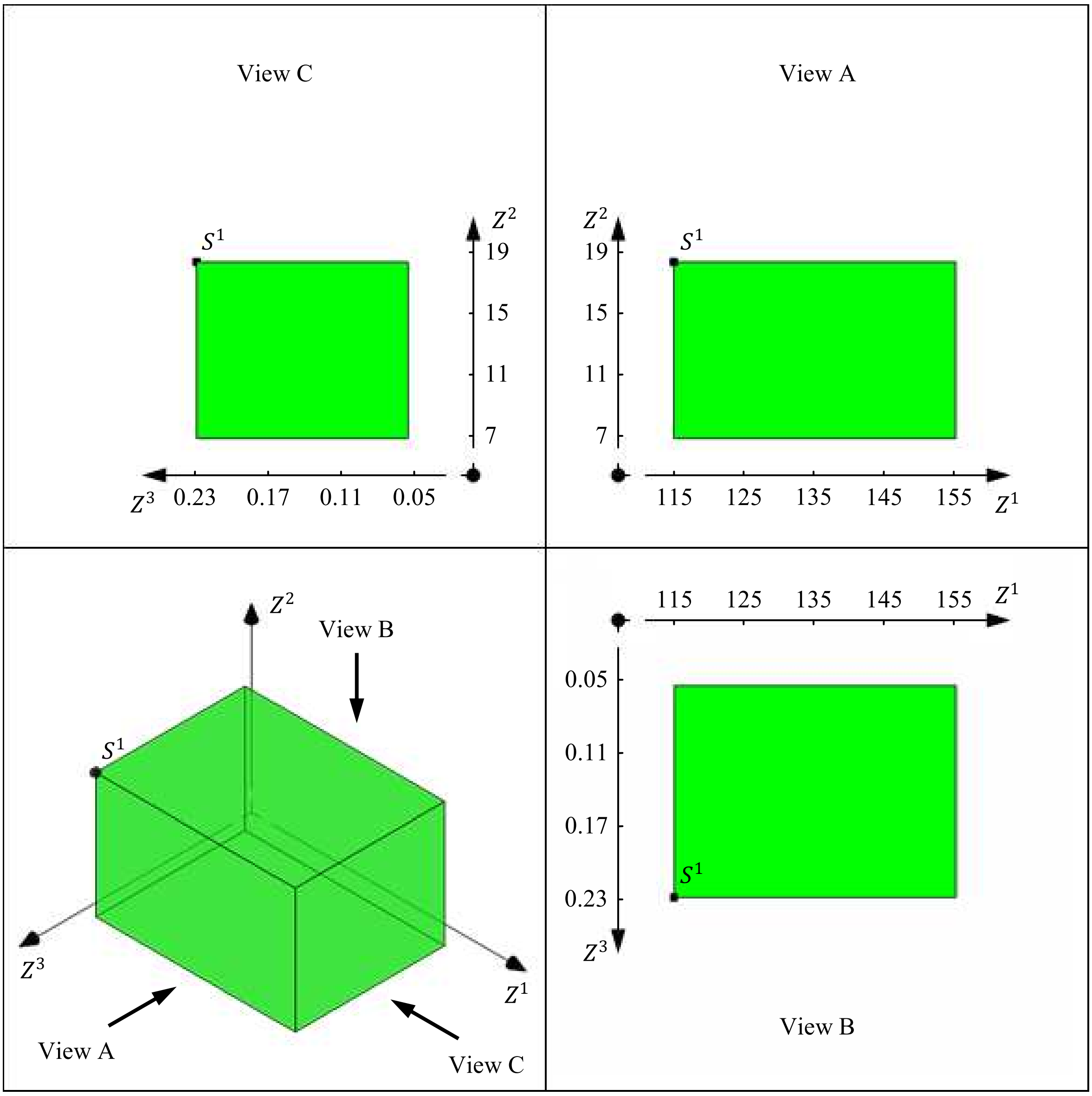


Figure 5.12 Region characterization of the method after the inclusion of S^1 (test instance).

Thus, the region characterization seen in Figure 5.15 is the first image to be shown to the DM, which already has some areas in the objectives space explored. From this point forward, human

(DM) interaction is required at each iteration in order to continue the search for non-dominated solutions. The weighted sum program to be solved at each iteration is the following:

$$(\text{CLRP}_3) \quad \min \quad \lambda^1 Z^1 + \lambda^2 Z^2 + \lambda^3 Z^3 \tag{5.1}$$

$$\text{s.t.:} \quad x \in X.$$

Where: the weights λ^i $(i = 1,\ 2,\ 3)$ satisfy $\sum_{i=1}^{3} \lambda^i = 1$ and $\lambda^i > 0$; Z^i $(i = 1,\ 2,\ 3)$ are the objective functions of CLRP$_2$ as given in (4.1), (4.2), and (4.3); and constraints are the same as defined for CLRP$_2$.

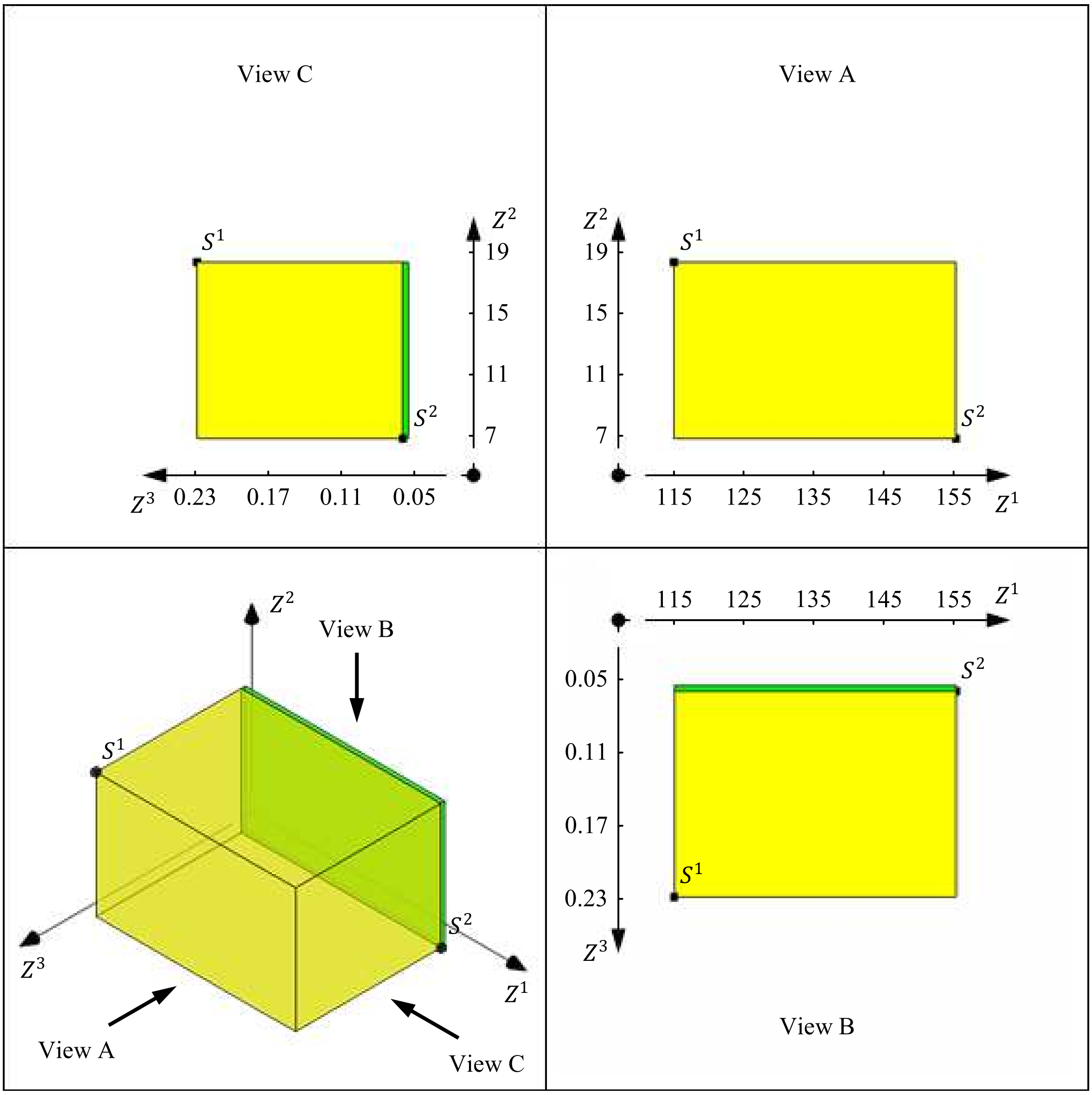


Figure 5.13 Region characterization of the method after the inclusion of S^2 (test instance).

Let us assume that, at this point (in the first iteration of the method), the DM wants to explore the green region, in order to find solutions that are better than any current solution with regards to two objective functions (possibly providing a more preferred trade-off).

Let ε denote a positive small enough number, based on the aforementioned DM's preferences (from which the region of interest is obtained), the following constraints are added to the $CLRP_3$ formulation:

$$Z^1 \leq 138.4380 - \varepsilon, \tag{5.2}$$

$$Z^2 \leq 7.9179 - \varepsilon, \tag{5.3}$$

$$Z^3 \leq 0.059434 - \varepsilon. \tag{5.4}$$

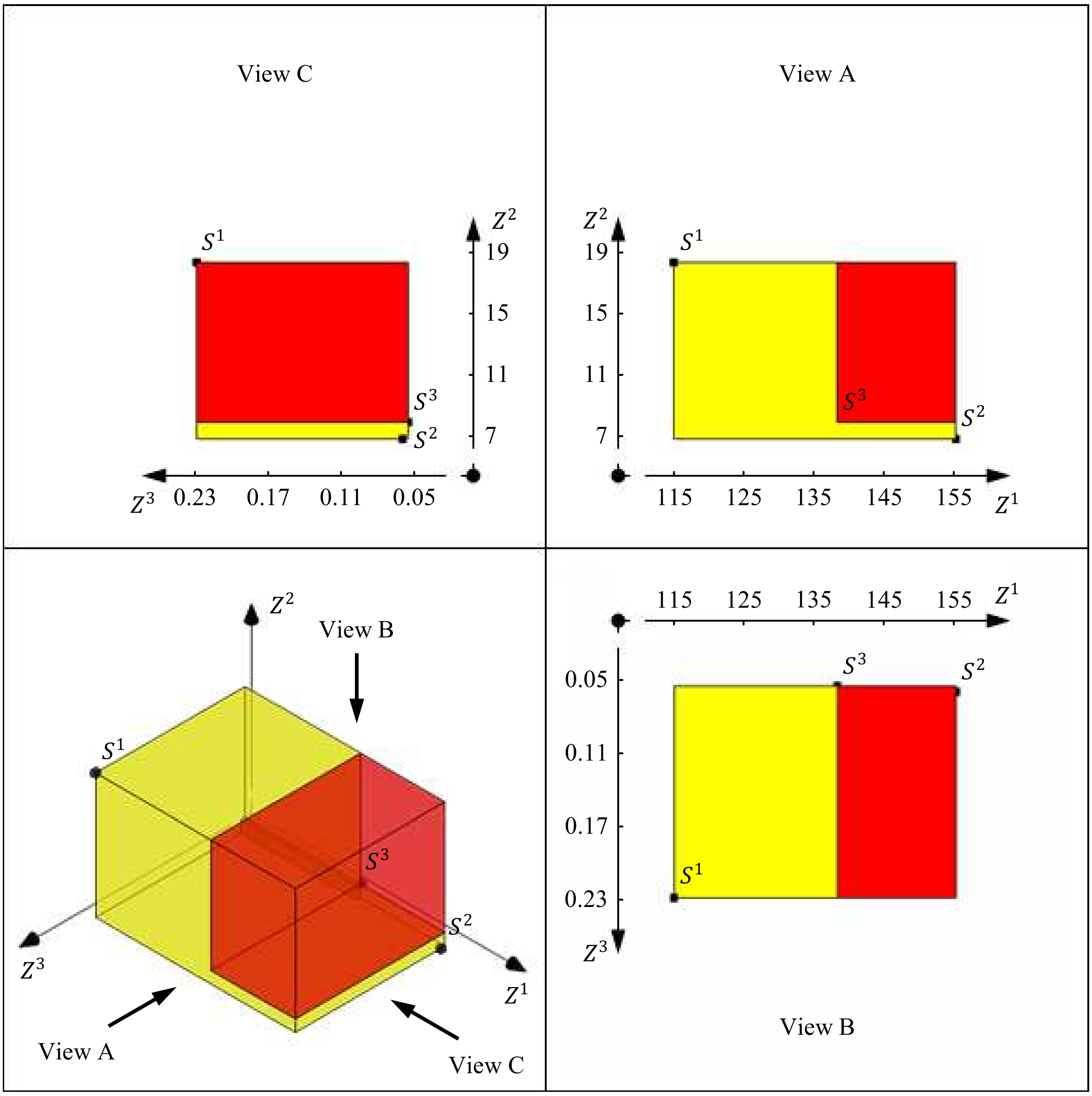


Figure 5.14 Region characterization of the method after the inclusion of S^3 (test instance).

Solving the new formulation finds the problem to be unfeasible. Based on this information, the region of interest is considered explored as is void of any non-dominated solutions (Figure 5.16).

In the second iteration, it is assumed that the DM is searching for a solution which, comparing with S^3, may be more expensive (worse regarding Z^1), but with an inferior overall obnoxious effect (better regarding Z^2). The corresponding region of interest can be seen in Figure 5.17 (subregion with colour blue), being the following constraints therefore added to the original CLRP$_3$ formulation (with $\varepsilon > 0$ and small enough):

$$Z^1 \geq 138.4380 + \varepsilon, \tag{5.5}$$

$$Z^2 \leq 7.9179 - \varepsilon. \tag{5.6}$$

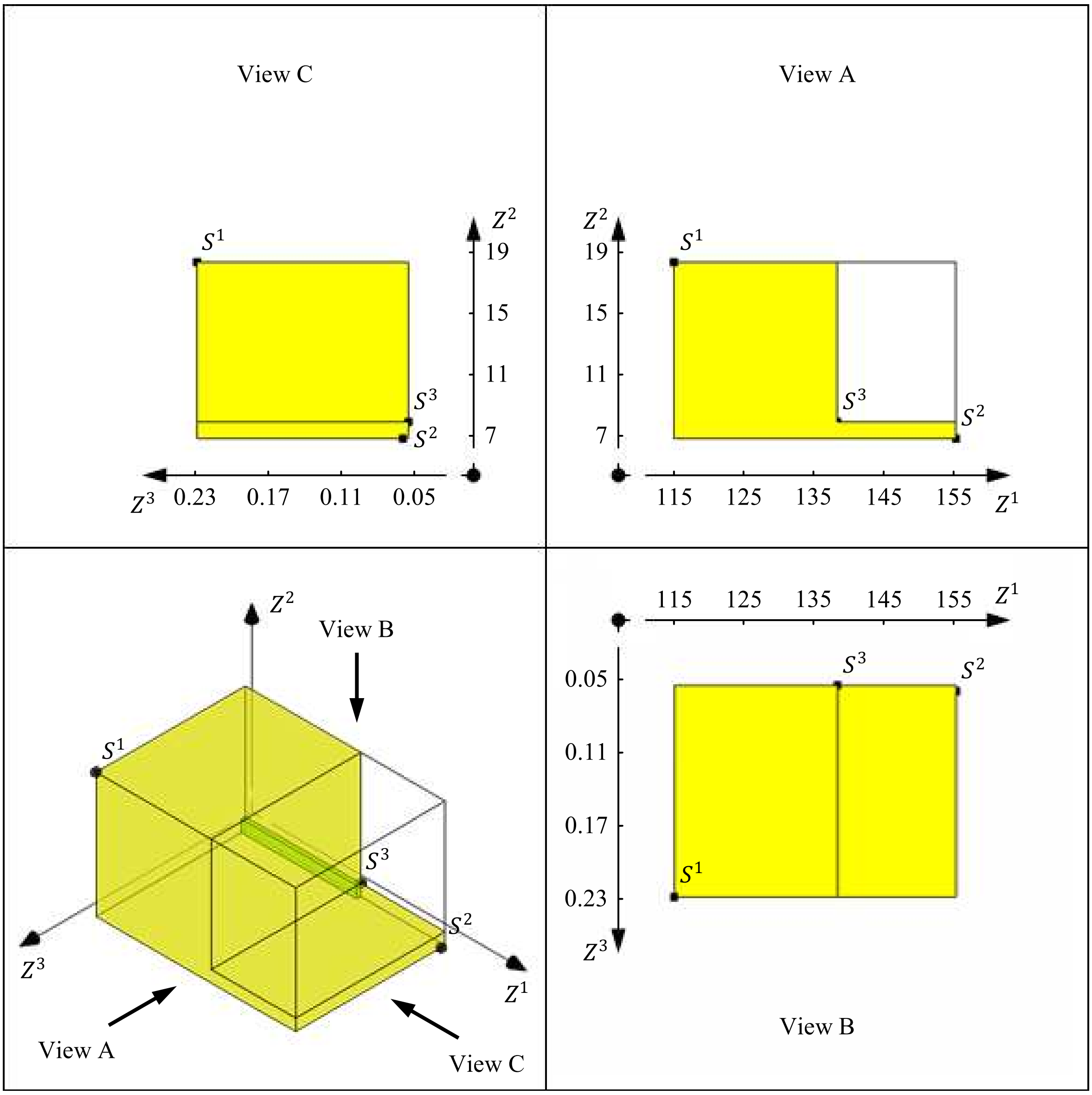


Figure 5.15 Output of the first step of the interactive method (test instance).

By solving the new formulation to optimality, $S^4 = (143.0114,\ 7.5433,\ 0.057692)$ is attained, allowing to characterize the following regions:

- coloured red, by unfeasibility, when $Z^1 \leq 143.0114 \cap Z^2 \leq 7.5433 \cap Z^3 \leq 0.057692$
- coloured red, by dominance, when $Z^1 \geq 143.0114 \cap Z^2 \geq 7.5433 \cap Z^3 \geq 0.057692$
- coloured yellow, when $Z^1 \leq 143.0114 \cap Z^2 \geq 7.5433 \cap Z^3 \geq 0.057692$
- coloured yellow, when $Z^1 \geq 143.0114 \cap Z^2 \leq 7.5433 \cap Z^3 \geq 0.057692$
- coloured yellow, when $Z^1 \geq 143.0114 \cap Z^2 \geq 7.5433 \cap Z^3 \leq 0.057692$.

Based on the output of the previous iteration and obeying to the colour hierarchy, Figure 5.18 is obtained.

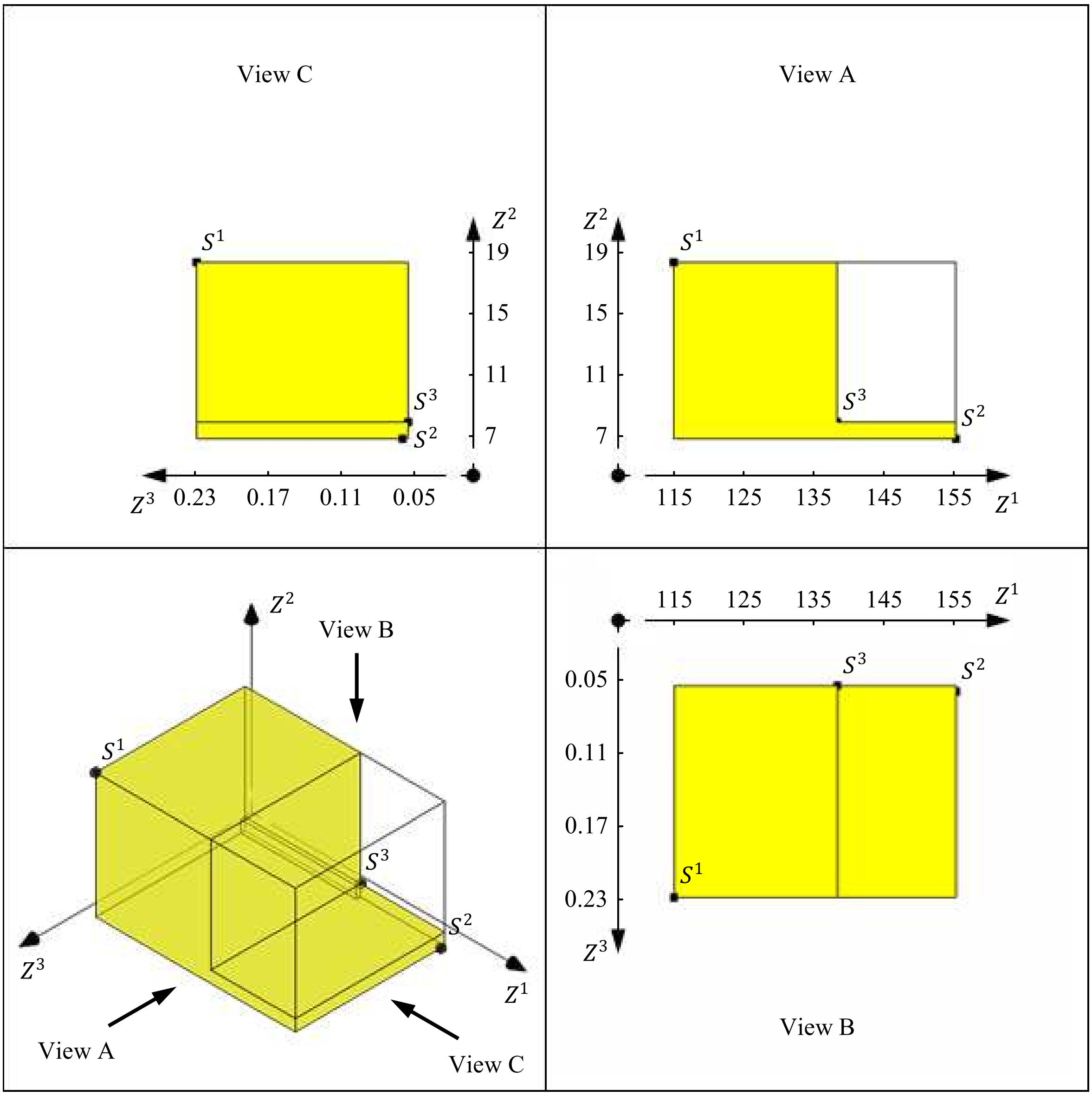


Figure 5.16 First iteration of the method after no solution was found in the region of interest (test instance).

Let us assumed that the DM considers not to have sufficient knowledge about the set of Pareto solutions. Thus, the method proceeds to a third iteration where it is assumed that the DM wants to search for a solution that has a good trade-off regarding both cost and equity (respectively, Z^1 and Z^3) while completely disregarding the overall obnoxious effect (Z^2). To that extent, the DM defines a region of interest with an upper bound for Z^1 of 120 and for Z^3 of 0.1 (as seen in Figure 5.19). Likewise, to the original CLRP$_3$ formulation, the following upper bounds are added:

$$Z^1 \leq 120, \tag{5.7}$$

$$Z^3 \leq 0.1. \tag{5.8}$$

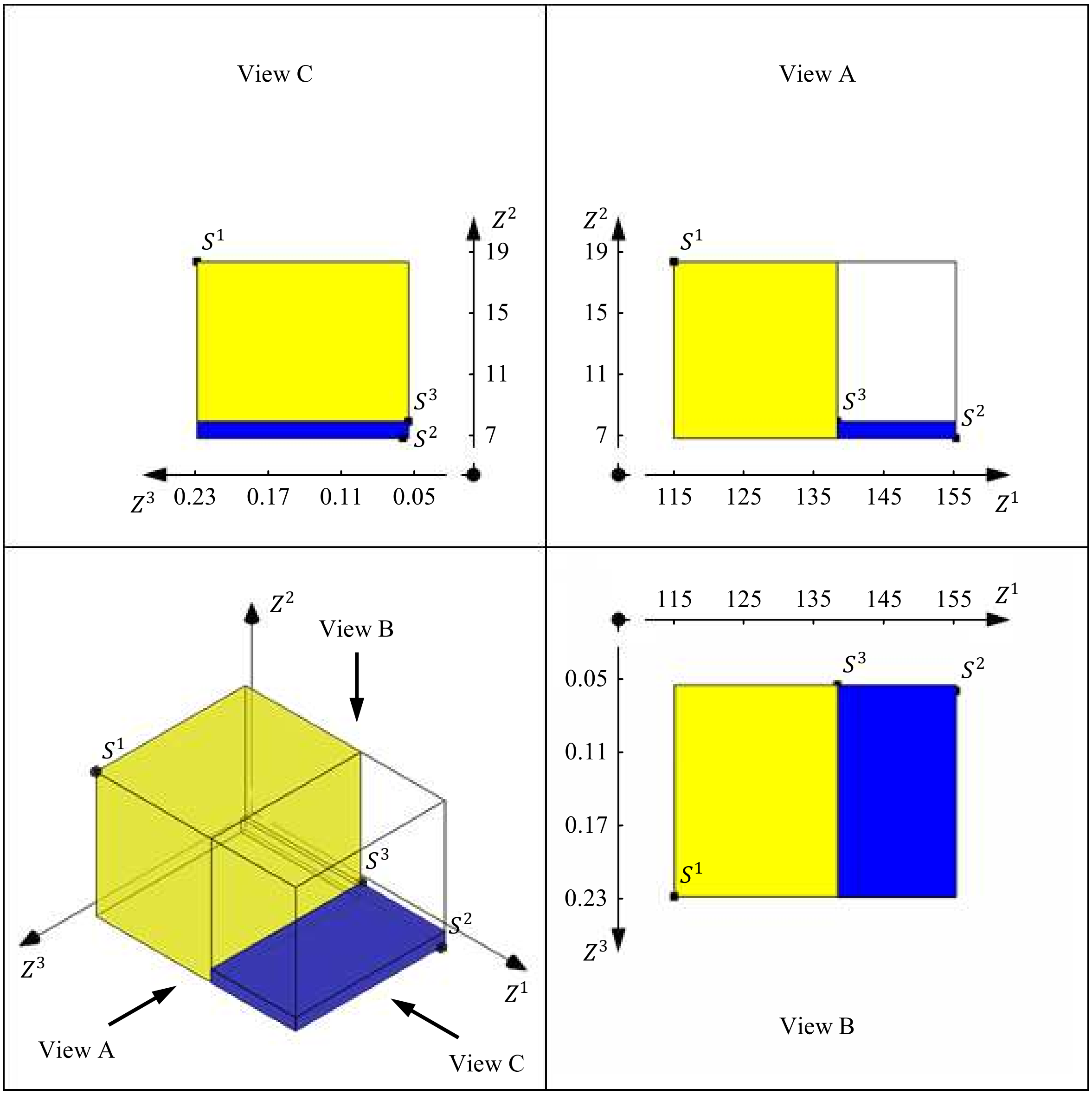


Figure 5.17 Second iteration of the method with the DM-defined region of interest (test instance).

The new formulation, when solved to optimality, is able to obtain a new non-dominated solution, say S^5, with $S^5 = (117.0605,\ 11.6599,\ 0.059004)$. Again, this allows to characterize several regions in the following fashion:

- coloured red, by unfeasibility, when $Z^1 \leq 117.0605 \cap Z^2 \leq 11.6599 \cap Z^3 \leq 0.059004$
- coloured red, by dominance, when $Z^1 \geq 117.0605 \cap Z^2 \geq 11.6599 \cap Z^3 \geq 0.059004$
- coloured yellow, when $Z^1 \leq 117.0605 \cap Z^2 \geq 11.6599 \cap Z^3 \geq 0.059004$
- coloured yellow, when $Z^1 \geq 117.0605 \cap Z^2 \leq 11.6599 \cap Z^3 \geq 0.059004$
- coloured yellow, when $Z^1 \geq 117.0605 \cap Z^2 \geq 11.6599 \cap Z^3 \leq 0.059004$.

This region characterization (obeying to the colour hierarchy) can be seen in Figure 5.20.

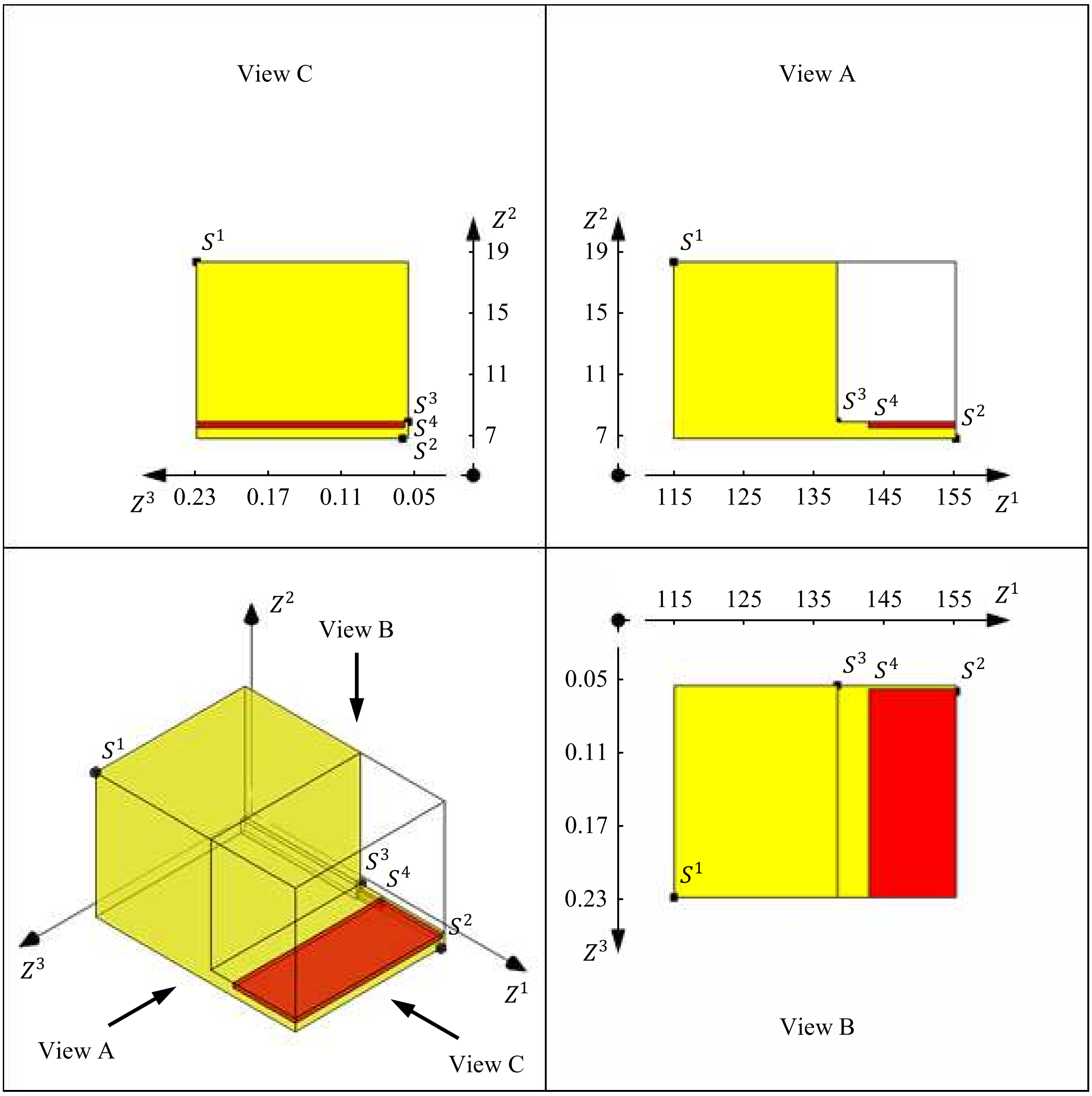


Figure 5.18 Second iteration of the method with the region characterization after obtaining S^4 (test instance).

S^5 is a highly equitable solution, while at the same time being only 1.77% worse than the ideal value regarding cost (Z^1). As it seems to be an interesting trade-off to this specific instance, it is assumed that the DM is satisfied with her/his current knowledge on the set of non-dominated solutions, and thus the method ends. Were it not the case, new iterations would follow (similarly to the previous three shown), ultimately ending the method when the DM would so desire, or all of the objectives space had been explored. Figure 5.21 depicts the graphical representation of the efficient solutions S^4 and S^5.

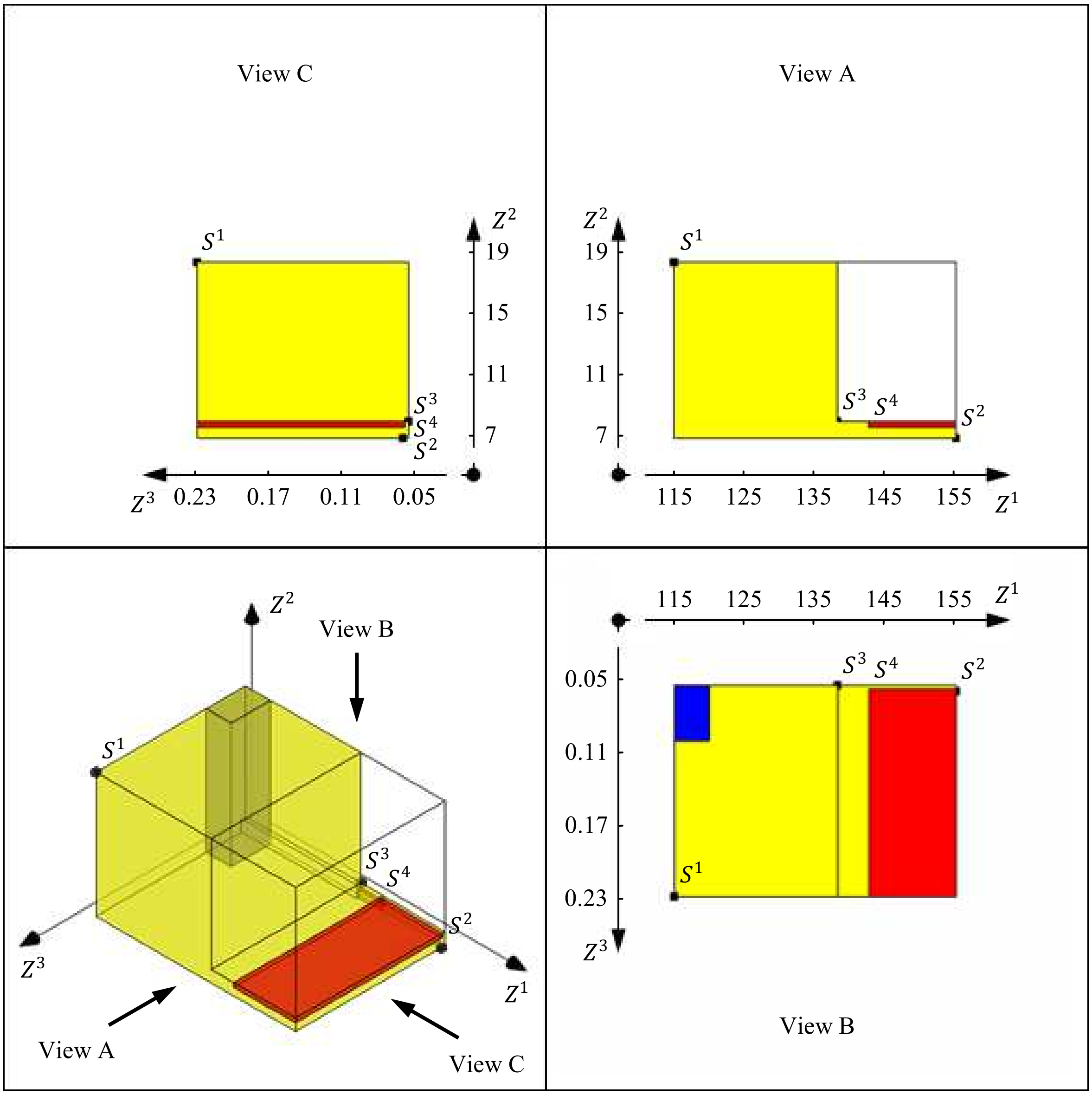


Figure 5.19 Third iteration of the method with the DM-defined region of interest (test instance).

In Appendix C all the non-dominated solutions as well as the graphical representation of the corresponding efficient solutions of this test instance can be seen. Looking at the results, some conclusions can be drawn. All solutions have 3 routes, which is due to routing not being obnoxious, leading to solutions with fewer vehicles in order to obtain the least feasible cost (Z^1) for each combination of depots to install. Also, the best solution regarding cost (S^1) is the only one that requires to install three depots (one per route). The remaining solutions, in order to reduce the (individual and overall) obnoxious effect, require to install the least feasible number of depots (due to capacity constraints): two.

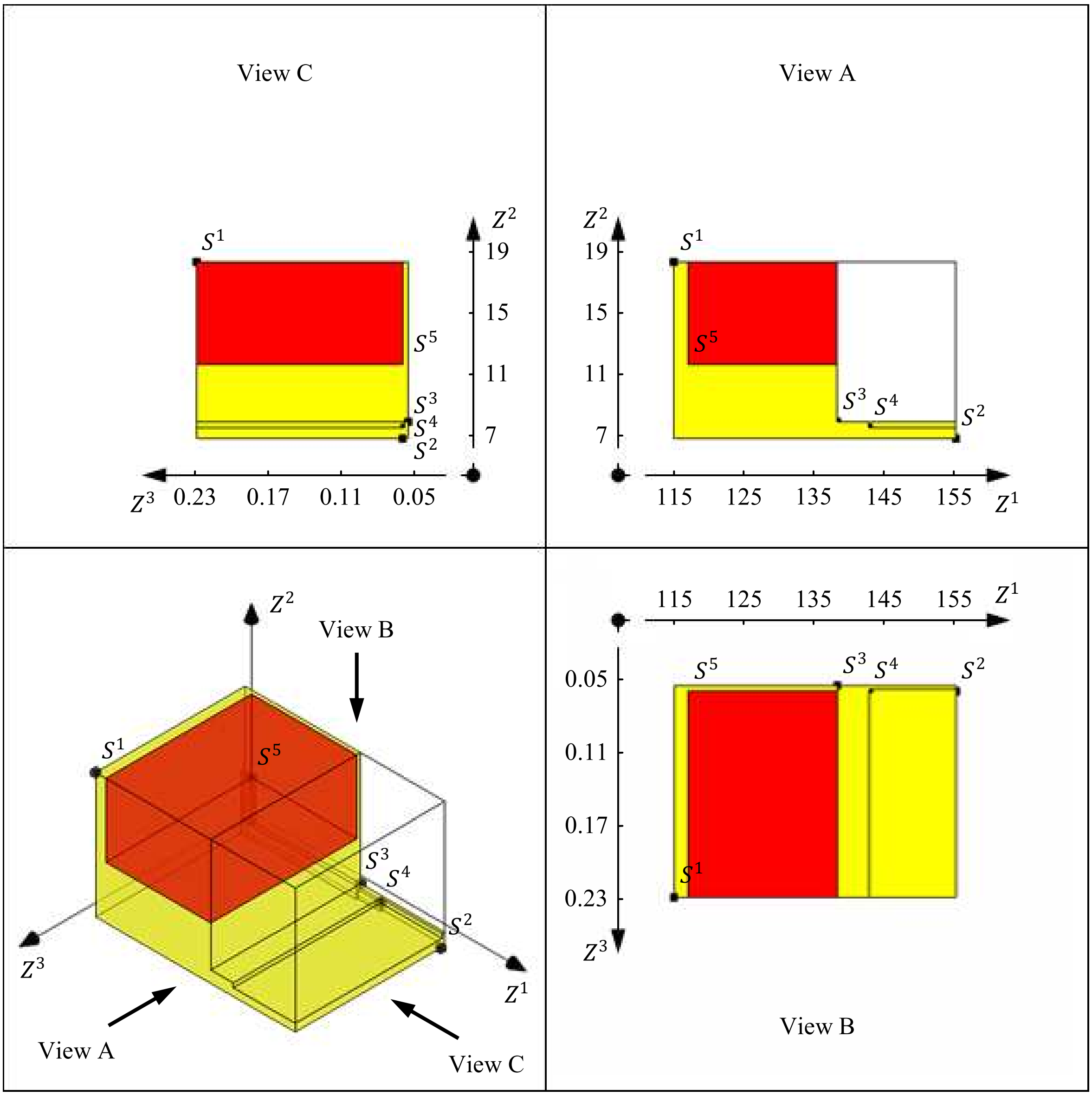


Figure 5.20 Third iteration of the method with the region characterization after obtaining S^5 (test instance).

Other conclusions (drawn in Section 4.3.3) are also confirmed: obnoxious effect and equity seem positively correlated, while cost is negatively correlated with the remaining two objectives; although the tracing of some routes is common to more than one solution (eventually only varying the links to the depots to install), significantly different route configurations can be found, validating the use of integrated location-routing multi-objective approaches.

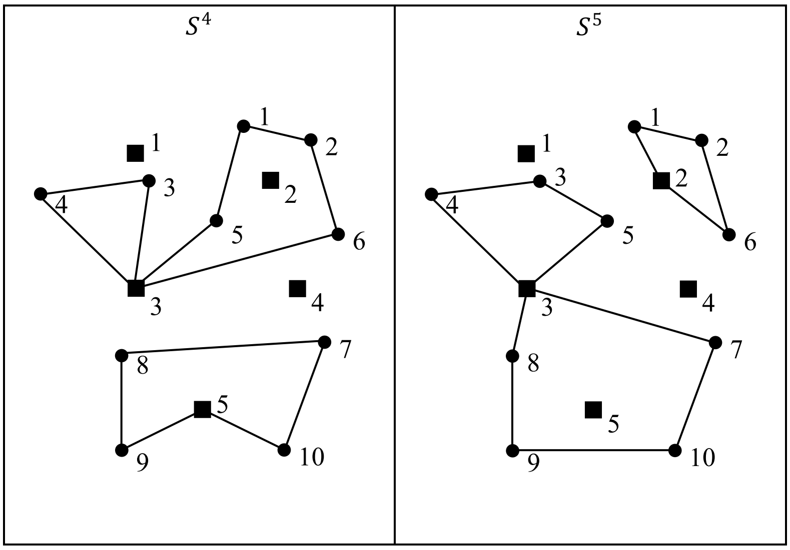


Figure 5.21 Graphical representation of the efficient solutions S^4 and S^5 of the test instance.

5.4 Summary

In this chapter an interactive method for MOMIP is proposed. Firstly, MOMIP current approaches are introduced and discussed (also some multi-objective problem concepts are defined). Then, the work focuses on interactive methods, where there is an alternation between human intervention and computation phases. For these methods two different paradigms exist, the use of an implicit utility function and an open communication protocol.

Given the characteristics of both paradigms, the use of open communication methods seems to be the most advantageous (for the required application). Thus, methods using this paradigm are reviewed, and a new open communication interactive method is presented.

The newly proposed method is based on the method by Ferreira et al. (1996), extending it to the multi-objective case (as it was restricted to problems with two objective functions). The proposed method relies on graphical and numerical information, not being too demanding from the DM point of view (possibly even allowing to unfold data in a way that makes it visually appealing). Computation wise, it uses a weighted sum program that enables to obtain all non-dominated solutions of a given problem.

A step-by-step example, applied to a multi-objective CLRP, is then provided, allowing to further examine the workings of the proposed interactive method.

Chapter 6

Development of Decision Support Systems

Traditionally, the decision-making process required a lot of experience on either the addressed problem and/or the methods/approaches to solve it. Nowadays, computation tools are able to decrease significantly both the need for acquired experience, as well as the time to obtain solutions for a specific scenario. A computation tool intended to support managerial decisions is called a decision support system (DSS).

In this chapter, the development of DSSs is addressed. Firstly, an introduction to these information systems is made, where the applicability to the location-routing problem (LRP) is studied. Secondly, as the development of a DSS follows the development of any other information system, current main software development methodologies are briefly reviewed, and human-computer interaction (HCI) issues discussed. Finally, the focus will be on a decision-support tool for the LRP, where the main preliminary phases of the adopted software development process are presented (exploration, planning, and iterations to release).

6.1 Introduction to Decision Support Systems

Decision support frameworks can be divided according to the degree of structuredness, ranging from highly structured to highly unstructured decisions/problems, and the type of decision, which can be strategic, tactical or operational (Gorry and Scott-Morton, 1971).

The degree of structuredness is based on the decision-making process by Simon (1977) composed of four phases (initially three, and later added the implementation phase): intelligence (involves searching for conditions that call for a decision), design (involves developing and analysing possible alternative courses of action), choice (where a course of action is selected from among the available ones), and implementation (involves adapting the selected course of action to the decision situation).

In structured problems (where all phases of the decision-making process are structured) procedures to obtain solution(s) are known and all aspects of the problems are described with a high degree of completeness. The problem can thus be broken down into a series of well-defined steps, leading to solutions which the decision maker (DM) can easily agree on. Unstructured problems are the opposite (with none of the four phases being structured). As they cannot be solved with a high degree of certainty and only aspects of the problem are considered, DMs often disagree about the best solution, requiring the use of intuition, reasoning, and memory. Semi-structured problems fall between structured and unstructured problems, having some elements of both. LRPs can be considered as semi-structured as, although methods to obtain solutions are known, the problem itself is not fully described (often oversimplified), thus relying on the DM's experience for the choice of which solution to implement.

Solving semi-structured problems may involve a combination of standard solution procedures and human judgement (Figure 6.1). For the structured portion of the decision, (operations research) models can be used; for the unstructured portion, a DSS can improve the quality of the information provided to the DM (e.g. by providing not only one but several alternative solutions and their potential impacts). This may help to better understand the nature of problems and thus to make better decisions (Turban et al., 2007). Moreover, DSSs are able to relax cognitive, temporal and/or economic limits on the DM, making them instrumental in decision-making scenarios (Holsapple, 2008).

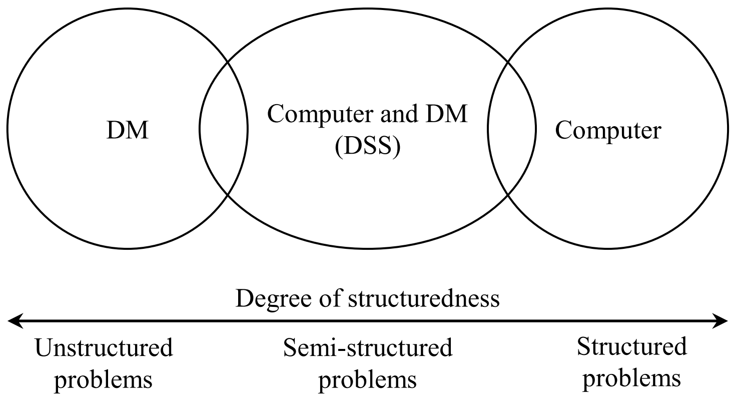


Figure 6.1 Decision making according to the degree of structuredness.

DSSs comprise a core subject area within the information systems discipline where business and organizational decision-making activities are supported (Burstein and Holsapple, 2008). Gorry and Scott-Morton (1971) originally define DSS as an interactive computer-based system which helps DMs use data and models to solve semi-structured and unstructured problems. Keen and Scott Morton (1978) present a similar definition, stating DSSs couple intellectual resources with computer capabilities in order to improve the quality of decisions in semi-structured problems. These are classic definitions, which have been extended over time. For example, Turban et al. (2007) define DSS as an umbrella term to describe any computerized system that supports decision making inside an organization.

Here, the term DSS will be used following the definition by Turban et al. (2007), meaning any computerized system developed to support decisions regarding unstructured or semi-structured problems, using development processes and concepts of information systems. The application itself will be named decision-support tool (DST).

The fundamental components of the DSS architecture are (Marakas, 1998): the data, the model(s), and the user interface. Data (which can come from many sources) are needed to solve problems. Any problem to be solved, opportunity or strategy to be analysed requires some data. The data (of a specific scenario) are then manipulated by using models, which can be standard (e.g. spreadsheet functions) or customized (e.g. an algorithm to solve a specific problem). Models can be used to forecast the possible outcomes of decisions. As the DSS requires interaction with users, the user interface is crucial for a successful system, as it enables users to use, support or examine data, and allows them to analyse and evaluate solutions obtained by models. Figure 6.2 depicts how these main components interact.

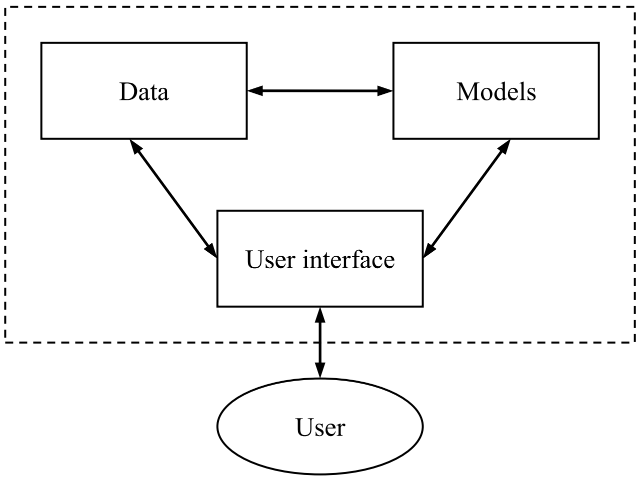


Figure 6.2 Main components of DSSs architecture.

The development of DSS applications (DSTs) have been fairly active for several areas (see Eom and Lee, 1990, Eom et al., 1998, and Eom and Kim, 2006, for surveys of DSTs). However, for LRPs only few works have surfaced (the works by Coutinho-Rodrigues et al., 1997, Gorr et al., 2001, and Lopes et al., 2008a), all of them restricted to specific models (e.g. the single-objective capacitated LRP in the work by Lopes et al., 2008a). The LRP, however, has several applications and variants (see Chapter 2), motivating the need to develop new DSTs capable of addressing (possibly several of) these problems. Moreover, the development and availability of DSTs may help both DMs and researchers. The former, by allowing to (easily) obtain scientifically-supported solutions may improve the quality of decisions. The latter, by aiding the process of gathering data, as well as enabling to visualize the inner-workings of developed models, reduces the time to obtain (real-world) instances and improves the functioning of models making them easier to understand and improve (e.g. in the tuning of parameters).

The development of such a DST (solver-oriented, according to the classification by Holsapple, 2008) should follow the development steps of any other information system software. Thus, in the next sections, the development process of DSSs will be reviewed (identifying main strengths and weakness of different software development methodologies), HCI issues discussed, and the steps adopted for the development of a DST for LRPs presented.

6.2 Development Process of Decision Support Systems

The development of a DSS usually requires the involvement of several interested parties, to develop the required components (data, models, and user interface), throughout the different activities that compose the development process (Figure 6.3).

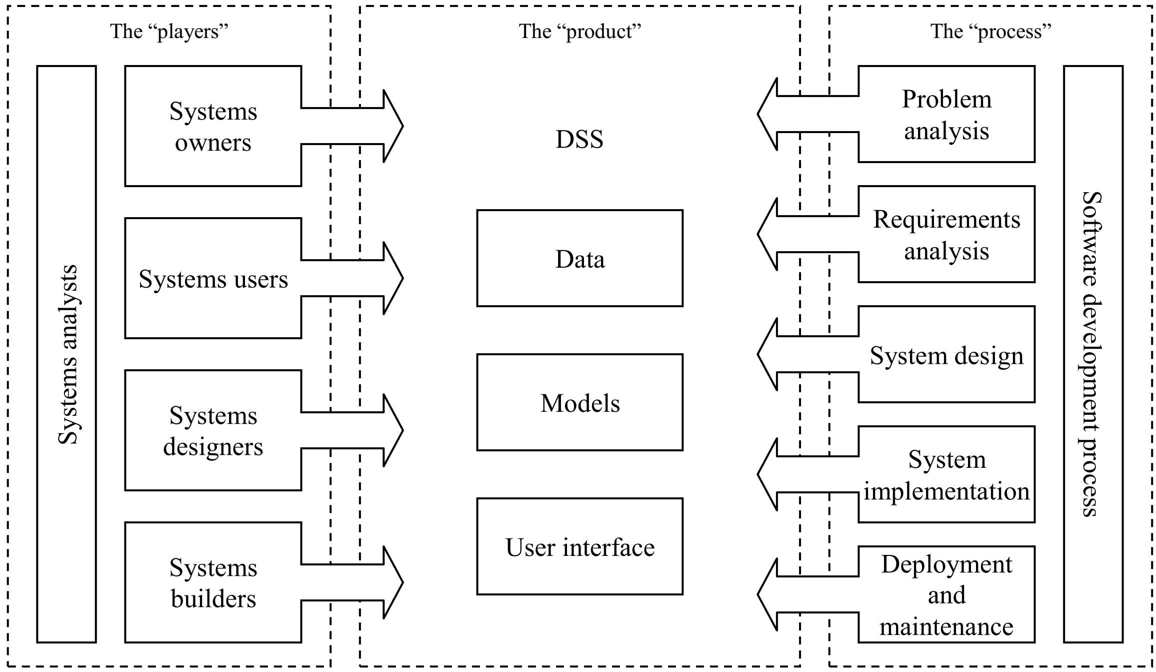


Figure 6.3 Development process of DSSs (Whitten and Bentley, 2007).

The parties interested in the development of DSSs are named stakeholders and can be both technical and non-technical. Within the technical stakeholders, systems analysts, systems designers, and systems builders can be included. Non-technical stakeholders (which are not part of the project's technical teams) also play an important role, being systems owners and systems users.

According to Whitten and Bentley (2007), their main roles are the following (left-hand side of Figure 6.3):

- Systems analysts – coordinate the efforts of the remaining stakeholders during the development process. They serve as facilitators, bridging the communication gap that often exists between technical and non-technical stakeholders.
- Systems owners – usually managers which tend to be more interested in the cost of the system, return value or benefits it will bring to the business. They pay for the system to be built and maintained, being among their biggest beneficiaries. Their contribution is important as they hold the knowledge of the organization's business and mission.
- Systems users – someone which will use or is affected by the system on a regular basis. Unlike system owners, they are mostly concerned with the functionality the system provides

to their jobs, and the system's ease of use and learning. Being the main target of systems, they play a key role in the correct identification of requirements.

- Systems designers – technological specialists interested in the correct technological choices and in the design of systems that support them. These can be database administrators (which design and coordinate databases), network architects (specialized in designing, installing, configuring and optimizing networks), Web architects (specialist who design complex Web sites) or even graphic artists (to design and construct compelling and easy-to-use interfaces).
- Systems builders – construct the system according to the systems designers' specification. These are mostly programmers, being specialists who convert requirements and statement of problems into computer applications, thus developing, testing and implementing systems.

Depending on the size of the system (and often in practice), any one individual may play more than one of the different aforementioned roles.

Regarding the main components of DSSs (which have been previously addressed), the choice of the most appropriate technologies varies according to the intended decision-support scenario/environment.

Finally, concerning the development process of DSSs (right-hand side of Figure 6.3), several activities are required. In the following subsections the different main activities will be presented, followed by a review of current main software development methodologies and HCI issues.

6.2.1 Main Development Activities

The development process of any DSS (as of any information system) is characterized by several activities. Tasks considered within are, usually, identifying the problem, its analysis and understanding, identifying requirements, designing the solution and then, encoding, testing, deploying and maintaining the designed solutions.

Overall, these tasks can be grouped into five main activities (Teixeira, 2008):

- Problem analysis – where usually the systems analysts together with the owners establish the project scope, goals, schedule, and budget required to solve the problem (or found opportunity).
- Requirements analysis – the main goal is to provide a more thorough understanding of the problems and needs that triggered the project. This is done by system analysts, which define with system users their needs with regards to functional and non-functional requirements (these will be further described in Section 6.3.1).
- System design – refers to an abstract representation of the system where requirements are converted into a modelling language, in order to facilitate communication between stakeholders (namely, systems builders, which will then implement the solution). The primary stakeholders in this activity are systems analysts and systems designers.
- System implementation – where systems builders (often programmers) have to interpret the models obtained in the system design activity, and convert them into programming language (encoding the models) to be executed in computers. Here, the system is constructed and

tested (preferably by an individual other than the programmer that encoded it). This activity should involve systems analysts and systems builders.

- Deployment and maintenance – in this activity, the system is made available to systems users and placed into operation. Afterwards, the system will eventually need to be changed, be it due to found errors (either from software "bugs" or design and implementation flaws) or the need to continuously improve/adapt the system.

These can be seen as the main activities in the development of DSSs which, over the years, have been used in different development processes. The different processes have been studied in an area named software engineering (Pressman, 2001; Sommerville, 2007), which concerns the practicalities of developing and delivering useful software (here, the software being the DST). These development processes usually follow a methodology. In the next section, some of the most known and used software development methodologies will be briefly reviewed.

6.2.2 A Brief Review of Main Software Development Methodologies

A software development process (or software lifecycle) is an approach that structures the development of a software product. The process defines not only the sequence in which the different tasks are to be performed but also the stakeholders involved and how to reach a certain goal (the "what", "when", "who", and "how").

There are currently several methodologies for the software development process, each describing different approaches to the main development activities and tasks within. Some of the main methodologies in the literature will be briefly reviewed in the following subsections, where an overview, corresponding advantages and disadvantages are provided. From the reviewed methodologies only the first adopts a linear framework (where main activities are performed sequentially until the last phase is achieved and the methodology ends), the remaining methodologies use an iterative and incremental framework (in which development is incremental with activities being performed in iterations, often cyclic).

Waterfall

The waterfall methodology is a sequential development process (linear framework) in which the phases correspond to the main development activities, and are performed flowing downwards (similarly to a waterfall, see Figure 6.4) without ever returning to a previous phase. In practice, the phases overlap and feed information to each other (Sommerville, 2007). At the end of each phase, tangible deliverables are produced and carried on to the next phase as inputs.

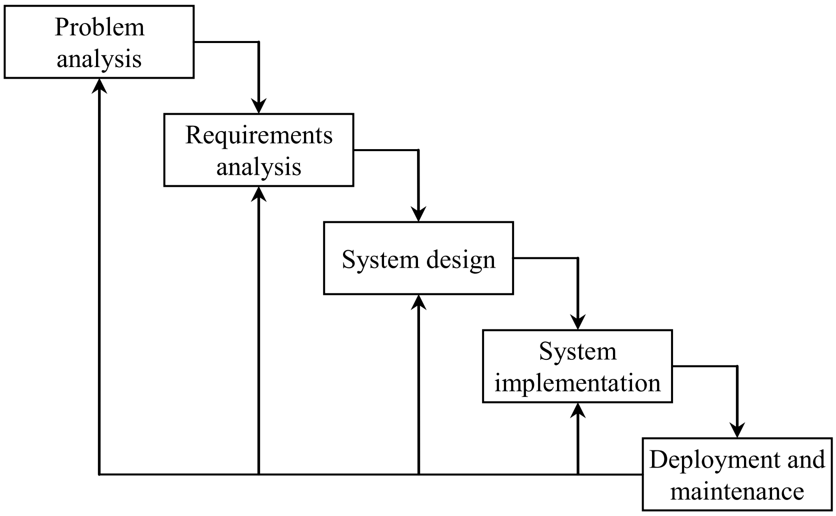


Figure 6.4 Software development process using the waterfall methodology.

The main advantage of this methodology is that the main focus is a clear definition of requirements, allowing better scheduling and (cost and time) estimating. Additionally, to fix and detect possible issues in early phases is often cheaper in time spent, money, and effort than fixing the same issues found later on in the development process (McConnell, 1996).

However, the simplicity of managing the process becomes increasingly complex as the size of the (software) project increases, making this methodology more suitable for smaller projects; as large projects have to be fully analysed and understood before proceeding to the following phases (which can be very time consuming). Also, the need to follow a set of rigid procedures is an obstacle to flexibility and change, which is often required in the development of some projects. Moreover, as it does not allow (during the development) feedback from the systems owners or users, the risk of the project not fitting the stakeholders' needs increases. A major part of this risk appears or rises towards the end of the project, with the cost of rectifying found issues usually increasing accordingly (Pressman, 2001).

This methodology may be more appropriate for smaller projects with a short lifespan, albeit its use in practice has been reducing in favour of more flexible methodologies.

Rapid Application Development

Rapid application development (RAD) is a methodology that uses an iterative and incremental framework, which emphasizes in obtaining short development cycles (Pressman, 2001). RAD uses minimum planning in favour of rapid prototyping (see Figure 6.5). The absence of extensive planning usually allows to develop software much faster.

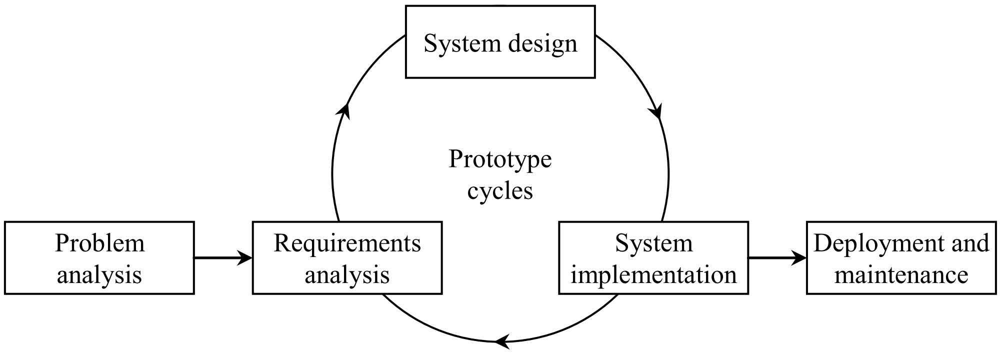


Figure 6.5 Software development process using RAD.

The biggest emphasis is on fulfilling the main requirements, while high performance is of lesser importance. This causes to not being recommended in the development of large projects. As it focuses on developing prototypes, that are iteratively developed into full applications, the software may lack the scalability of a solution that was designed from the start as a full application (Pressman, 2001).

Moreover, delivery deadlines are strict (timeboxing). If the project starts to fall behind in the schedule, the emphasis is on reducing the requirements in order to comply with deadlines. This causes applications to be less full featured than when using other methodologies (McConnell, 1996).

Rational Unified Process

The rational unified process (RUP), created by the Rational Software Corporation, uses an iterative and incremental framework while still inheriting some aspects of the linear framework. In RUP, feedback exists throughout the development process (although not often). The methodology requires the delivery of several artefacts throughout the development, thoroughly detailing the process.

RUP is divided into four development phases (Jacobson et al., 1999):

- Inception – where an outline of the project is drawn (and roughly estimated), major risks are identified and prioritized, and the elaboration phase is planned in detail.
- Elaboration – the system architecture is designed. Although the emphasis is on the requirements analysis and system design activities, implementation is started. At the end of this phase it is possible to plan the activities and estimate the resources required to complete the project.
- Construction – is when the software is built, thus implementation is the most prominent activity. In this phase the bulk of the required resources are expended, producing the first external release of the software.

- Transition – where the software is transitioned from development into production, making it available to users. Other activities include training users and testing the software to validate it against the users' expectations.

Figure 6.6 depicts the four development phases and their main activities (left-hand column). The curves in the figure approximate the extent to which corresponding activities are performed in each of the development phases.

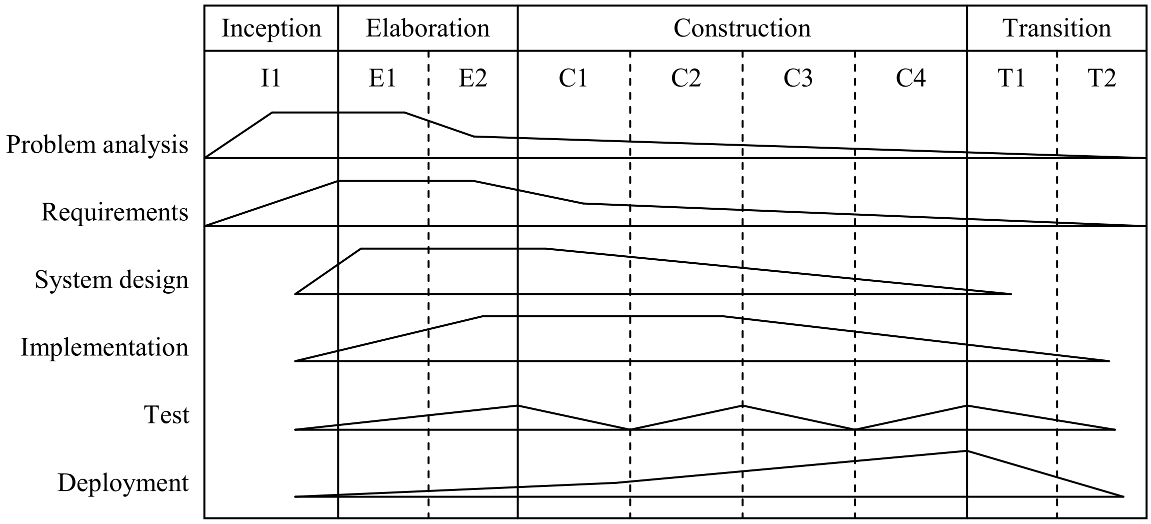


Figure 6.6 Software development process using RUP (Jacobson et al., 1999).

In each of the development phases, iterations are performed (e.g. in Figure 6.6, C3 is the third iteration of the construction phase) within which all main activities are usually performed.

This methodology is mostly oriented at large projects, where large development teams are involved, and all activities ought to be thoroughly documented. RUP overcomes some of the obstacles of the waterfall methodology, while still maintaining some of its advantages (Teixeira, 2008).

Agile Software Development

The agile methodology encompasses several methods that intend to overcome the rigidness and excessive number of deliverables of previous methodologies. It uses an iterative and incremental framework, encouraging frequent inspection and adaptation of the development process, teamwork, and collaboration between self-organizing cross-functional teams.

The agile manifesto, where the methodology was first presented, defines a set of principles that should be prioritize (Highsmith and Cockburn, 2001):

- individuals and interactions over processes and tools
- working software over comprehensive documentation

- customer collaboration over contract negotiation
- responding to change over following a plan.

These priorities do not dismiss the importance of the second set of items (processes, tools, documentation, contracts, and plans), but rather focuses on the first. This methodology relies on short feedback loops (at most six months) between systems users/owners and designers/builders, where working software is presented and evaluated (being the main measure of progress). Within each of these loops a miniaturized version of an entire software development process is performed, with a release at the end.

The agile methodology is most effective in smaller projects, where a constant shift of requirements may exist, allowing to quickly respond to them. If applied in large projects, it would be difficult to assess the effort required at the beginning of the software development process, and the lack of emphasis on necessary analysis, designing and documentation could prove to a be a risk (Sommerville, 2007).

Some of the best known agile methods are: extreme programming (XP), scrum, crystal, and adaptive software development. Among these, XP (Beck, 1999) is probably the best known and most widely used (Sommerville, 2007). XP is based on the notion of frequent releases in short development iterations, enabling to quickly obtain feedback from users. By using short iterations it may be easier to eliminate risks in the project and respond to changing requirements.

XP can be divided into five development phases (Ambler, 2002; Beck and Andres, 2004):

- Exploration – includes the development of the architectural spike and of the initial user stories. The architectural spike intends to identify areas of maximum risk, to get started with estimating them correctly. A user story is a high-level requirement formulated as a small text in the language of the user, being used for the specification of requirements.
- Planning – is where planning for the iterations and releases is performed, defining priorities for the different requirements (obtained from user stories). The purpose is to schedule a date by which the smallest, most valuable set of user stories will be implemented.
- Iterations to release – encompasses the primary effort of the project, as is where major development occurs. In this phase, system design, implementation and testing occurs iteratively, and several of these iterations may exist for a single release.
- Productionizing – focuses on certifying that the software is ready to go into production. Since it is the phase where releases are incorporated into the final product, extensive testing as well as performance tuning are required.
- Maintenance – is the normal state of XP projects, encompassing the planning, iterations to release, and productionizing phases after the first release of the system. This phase also includes other tasks, such as the operation and support of the system.

Figure 6.7 depicts the tasks performed within the different main development phases of the XP methodology and how they relate.

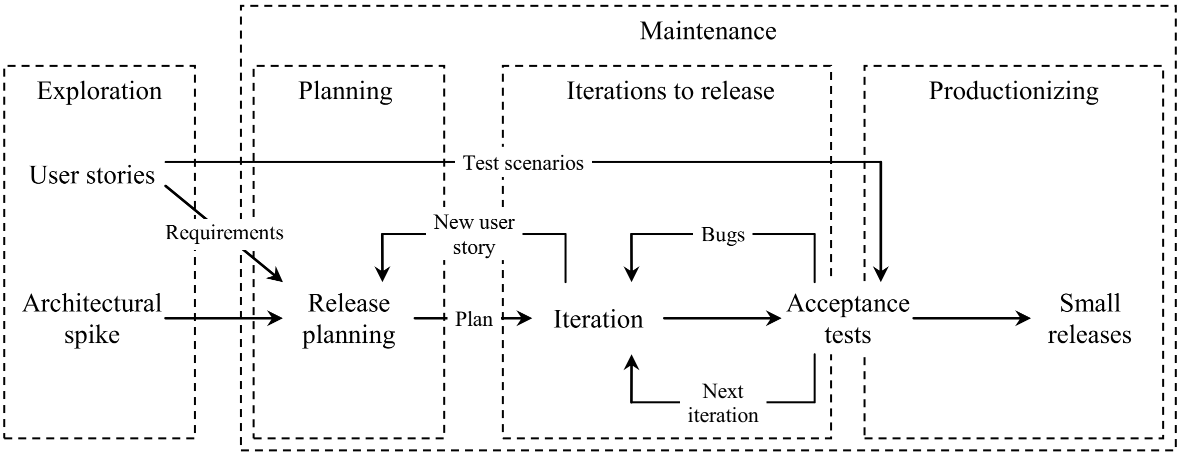


Figure 6.7 Software development process using XP (Ambler, 2002).

XP proposes a set of 12 different practices emphasizing four essential values (Beck and Andres, 2004): communication, simplicity, feedback, and courage. Constant communication with systems owners/users and fellow designers/builders is required. The design should be simple and clean. Feedback should be obtained by testing the software (with users) from the start, thus also allowing users to obtain feedback regarding its development. Facing the need to evaluate the software, users should have the courage to elicit new requirements and technical stakeholders should be able to courageously respond to changing requirements and technology.

6.2.3 A Brief Review of Relevant Human-Computer Interaction Issues

The development process (and final deployment) of DSSs may be more successful if the correct methodology is chosen, however, the overall success may depend mostly on the ability to interact with users (i.e. meet the user's expectations and needs, be it by allowing to easily attain goals, be more productive, or simply by providing an enjoyable experience). Most often it is that specific ability and not the number of functionalities or the overall quality that determines the outcome of a system (Sharp et al., 2007).

This aspect falls within the subject of HCI, where interaction between people (users) and computers is studied. This subject draws on many disciplines, although it is in computer science and system design that it must be considered as a central concern (Dix et al., 2004). The goal here is to study the interaction at the user interface software level (user interface design), in order to allow an easy interaction between humans and computers which, as seen previously, plays a key role in DSSs.

In order to achieve this, the concept of usability emerges. The standard ISO 9241-11 (ISO, 1998) defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specific context. Basically, usability refers to the study of the ease of use with which people can employ a particular tool in order to achieve a specific goal. Usability goals and objectives include (Rubin and Chisnell, 2008):

- Usefulness – concerns the degree to which a system enables a user to achieve her/his goals, being an assessment of the user's willingness to use the product.
- Efficiency – is the quickness with which the user's goal can be accurately and completely accomplished being usually a measure of time.
- Effectiveness – is the extent to which the system behaves in the way that users expect it to, and the ease with which they use it to do what they intend.
- Learnability – has to do with the user's ability to operate the system to some level of competence after some amount and period of training (which may be no time at all). It can also refer to the ability of infrequent users to relearn the system after periods of inactivity.
- Satisfaction – refers to the user's perceptions, feelings, and opinions of the system, usually captured through both written and oral questioning.

In order to specify or measure the usability of a system it is necessary to use techniques that allow to measure and verify the aforementioned goals. Usability engineering (Nielsen, 1993) is a field mainly concerned with HCI issues in human-computer interfaces. Currently, within usability engineering, several techniques exist (Dix et al., 2004; Rubin and Chisnell, 2008); some of these will be reviewed in the next subsections (an extensive survey and a taxonomy of usability evaluation techniques can be seen in Ivory and Hearst, 2001). Overall, they can be separated into techniques which require real (target) users and techniques which do not. The order in which they are described in the following subsections somewhat follows the order in which they are employed in the software development process. Among them, the most used is usability testing, mainly due to its cost/benefit ratio (Mitchell, 2007).

Paper Prototyping

Paper prototyping is a technique which requires real target users, as the appearance of the software is to be presented to them on paper, and questions regarding it are to be asked. This allows to learn, for example, whether the planned flow of screens (or pages) meets the users' expectations and needs. Questions made to users can range from navigation between screens to particular attributes, such as organization and layout, or even where certain options or types of information might be found.

Although it seems a simple technique it can provide a great deal of useful feedback. The value of paper prototype evaluation is that critical information can be collected quickly and inexpensively. It easily allows to determine which functions and features are intuitive and which are not. Moreover, users feel more comfortable being critical of paper prototypes because of not having a polished look (Klee, 2000).

Testing interfaces at early stages of the development process helps to identify software usability problems even before any code is written and development of the software begun. The costs and risks of later changes are therefore reduced and the overall quality of the software is increased.

Expert or Heuristic Evaluations

Expert evaluations involve reviewing a system, usually by usability specialists who have little or no involvement in the project. The specialists perform the review according to generally accepted usability principles (heuristics) and previous experience. Even though it does not require users, the specialists should adopt the view of the software's target users.

The heuristic evaluation is a technique that helps to identify interface usability problems. The most used usability heuristics for user interface design were proposed by Nielsen (1994). The set of heuristics is the following:

- visibility of system status
- match between system and real world
- user control and freedom
- consistency and standards
- error prevention
- recognition rather than recall
- flexibility and efficiency of use
- aesthetic and minimalist design
- help users recognize, diagnose, and recover from errors
- help and documentation.

According to Nielsen (1993), several different evaluators (usability specialists) should be used in order to find a significant number of usability problems. The author recommends five evaluators as a reasonable number (which may be able to find around 75% of usability problems). The evaluators should provide a list of usability problems, with ratings or judgments about the seriousness of the problems for users.

This technique is fairly inexpensive and can be used early in the development process, however it may not be able to identify as many usability issues as other techniques (e.g. usability testing). Moreover, results are often biased by the preconceptions of the evaluators.

Usability Testing

Usability testing usually involves selecting a group of users, representative of the system's target users, which are required to perform a set of realistic tasks in the system (thus being a technique which requires users). During the time that users perform the tasks, data regarding their performance are obtained (either using video or observation techniques). It is intended to evaluate the number of errors made by users, time required to perform the tasks, if they were able to successfully complete them, and if they chose the most adequate course of action. The behaviour of users should also be evaluated. Often a questionnaire is also provided to users in order to access their opinion of the system.

According to Rubin and Chisnell (2008), usability testing is most powerful and effective when implemented as part of an iterative development process. The authors stress that using this technique throughout the development process increases the probability of finishing with a usable system, as exposed usability deficiencies allow to gradually shape or mould it. Moreover, even if important flaws or deficiencies are missed during one of the tests, subsequent tests offer the opportunity to identify these problems or issues. According to the stage of the development process (and the corresponding main development activities) in which the tests are applied, Rubin and Chisnell (2008) classify them into:

- Exploratory or formative study – is conducted early in the development process (in requirements analysis) and, as such, the system is still in the preliminary stages of being defined and designed. The objective is to examine the effectiveness of preliminary design concepts.
- Assessment or summative test – is the most common type of usability test used, being conducted either early or midway into the development process (usually in system design). The purpose is to expand the findings of the exploratory study by evaluating the usability of lower-level operations and appearance of the system.
- Validation or verification test – usually conducted late in the development process (during system implementation activities) intending to measure the usability of the system against established benchmarks, or to confirm that problems found earlier have been solved and that new ones have not emerged.
- Comparison test – is not associated with any specific stage of the development process. In the early stages, it can be used to compare several radically different interface styles. Towards the middle, the test may allow to measure the effectiveness of a single element. Nearer the end of the development process, it may be used to see how the system stacks up against similar systems. This test is usually performed together with one of the previous (depending on when is performed).

Results of usability testing provide a reliable overview of the real problems that target users will encounter, helping to understand the users different behavioural patterns when using the system. This technique, however, is expensive, time consuming, and the validity of the findings relies heavily on the correct identification of target users and key tasks (Mitchell, 2007).

6.3 Development of a Decision-Support Tool for Location-Routing Problems

Previous sections have addressed methodologies and techniques for the development of DSSs. In this section the main phases of the methodology adopted for the development of a DST for LRPs will be presented, thus covering some of the main development activities. In Chapter 7, the developed tool (and its main components) will be presented and tested with regards to its usability (using usability testing as described earlier).

In order to develop the tool some concepts of the XP agile methodology were adopted (which will be discussed hereafter). This choice was due to the desire to make the tool available online and be able to quickly respond to user feedback, promoting a tool with continuous development (incorporating new models and functionalities, as well as improving the user interface), which may lead to a scenario of changing requirements and small releases.

Albeit some of the XP's practices may be perceived as inappropriate for research projects, Wood and Kleb (2002) have shown that, although some adaptations to the original practices have to be performed, several advantages could still be found with regards to more traditional methodologies. Likewise, some practices of the XP methodology were either not employed (e.g. pair programming and collective ownership) or had to be adapted (e.g. rather than an on-site customer, contact with users was performed during acceptance tests and in order to obtain user stories). Nevertheless, most of its defining practices were used, namely, planning game, simple design, the use of a naive metaphor, refactoring, unit tests, and the continuous and incremental incorporation of functionalities in small iterations.

The following subsections cover the first steps in the development of the DST following the main phases of the XP methodology: exploration, planning, and iterations to release. Within these phases, some of the main development activities of the software development process are also covered (such as requirements analysis, system design, and system implementation).

The exploration phase allowed to obtain the DST's current main high-level requirements (based on the user stories). Afterwards, in the planning phase, guidelines for planning iterations and releases are provided (in some cases high-level requirements are converted into a modelling language). Finally, in the iterations to release phase, mostly system implementation issues are addressed, where code is developed (based on the defined models, when available), tested, and integrated in the tool (after acceptance tests).

It should be stressed that development activities were (are to be) performed iteratively in short iterations, representing the main development phases (and providing best practices and guidelines to be) considered in (future) development iterations.

6.3.1 Exploration

In the exploration phase it is required to create spike solutions (in the architectural spike) to obtain answers to tough technical or design problems. A spike solution is a very simple program to explore potential solutions, where only the problem under examination is to be addressed (ignoring all other concerns). The goal is to reduce the risk of technical problems or increase the reliability of user stories estimations (Wells, 2009).

To that extent several small prototypes were developed in different programming languages, in order to test different objects and data input options, to meet some of the user stories. The final choice was on the extensible application markup language (XAML) technology, part of the Windows Presentation Foundation subsystem, with C# as code-behind. XAML is a declarative markup language (based on extensible markup language – XML), where objects and their properties are defined in XML. This choice was mainly due to its ability to separate the user interface from the code logic (allowing to easily change the interface at any given time without altering any of the implemented functionalities). Moreover, building user interfaces is easier and needs less code using XAML, while still maintaining the performance of the C# language (used for the code logic).

User stories are used for the specification of requirements and to create time estimates for the planning phase, being high-level requirements formulated as a short text in the language of the user (or written by the user). As it is essential to the correct development of the DST, definition of requirements has to be made. In this phase, only enough requirements to make a first good release are required (Beck and Andres, 2004). In the requirements analysis presented hereafter (although not strictly part of the XP methodology), requirements are classified and some user stories (high-level requirements) described, allowing to further understand the goals of the DST.

Requirements Analysis

Requirements analysis plays a key role in the success of a software. In software requirements engineering (Wiegers, 2003), a subdiscipline of software engineering, the underlying tasks are studied, namely, identifying stakeholders and their needs, determining the conditions to meet, and documenting all of them in the form of requirements that are easy to analyse, communicate, and implement.

A requirement is a property that must be exhibited. A software requirement is a property that must be exhibited by a software. Software requirements may be classified according to some of its attributes (Abran et al., 2004):

- High- and low-level requirements – high-level requirements are drawn from general functionalities, objectives, and business rules. Low-level requirements are based on the users' needs, functionalities and constraints.
- Priority rating – enables trade-offs when the project is subjected to finite resources. Allows specifying, for example, which requirements are mandatory and which are not.
- Product or process requirements – the first describe requirements on the software to be developed, while the second represent constraints on the development process of the software.
- Functional or non-functional requirements – this is the most widely used classification. Functional requirements specify the actions that a software must be able to perform (its functionalities). Non-functional (or quality) requirements are the ones that constraint the software by specifying properties (such as reliability, safety, performance, and usability). Non-functional requirements generally determine how the functional requirements will be implemented.

Requirements should be stated as clearly as possible (when possible, quantitatively) and should be verifiable. This is most important in non-functional requirements where goals should be defined in order to objectively test them (Abran et al., 2004).

User stories differ from traditional requirements specification in the level of detail. A user story should only have enough detail to make a reasonable estimate of the implementation time, providing high-level requirements to be used as input in the planning phase. The corresponding detailed requirements are to be obtained only when implementing the user story.

In the developed DST several user stories were obtained from system users. Some of them can be seen in Figure 6.8.

As a user, I can insert/edit data regarding the location problem.
As a user, I can obtain real online geographical data.
As a user, I can save all the data into a single file, allowing to load it later.
As a user, I can obtain several different solutions to location problems.
As a user, I can visualize on the map all the data regarding a solution.
As a user, I can print all the data regarding a solution.
Selected solutions are highlighted on the map.
As a user, I can insert data directly on the map.

Figure 6.8 Some user stories of the DST.

All the user stories assumed either the template form “as a [role], I can [capability]” (Cohn, 2004), or were expressed in the form of a story.

6.3.2 Planning

In the planning phase, the iterations and releases were scheduled. Implementation priorities of the high-level requirements (obtained from user stories) were defined, based on the perceived difficulty and the number of procedures to implement. For the correct scheduling, it was also necessary to predict, for each of the high-level requirements, the amount of time required to correctly implement it.

The highest priority requirements were implemented into several releases, each as a small project. The releases were separated into several iterations, each corresponding to the implementation of a high-level requirement (which in some cases corresponded to several functionalities), and subject to user acceptance.

In this phase, modeling is also a potential activity (thus, part of system design activities are performed in this phase). Following the principles of “model with a purpose” and “depict models simply”, only if there is a valid purpose should models be created and, when used, simple models should be adopted (Ambler, 2002). This need was identified in some of the most complex functionalities (e.g. obtaining solution, see Figure 6.9) for which models were created, in order to facilitate maintenance. For this purpose the unified modeling language (UML) was used.

UML is a standardized general-purpose modeling language in the field of software engineering, which can be used to describe software both structurally and behaviourally (Booch et al., 2005).

UML diagrams represent different views of the system: static (or structural) and dynamic (or behavioural). Within these different views, several types of diagrams can be found.

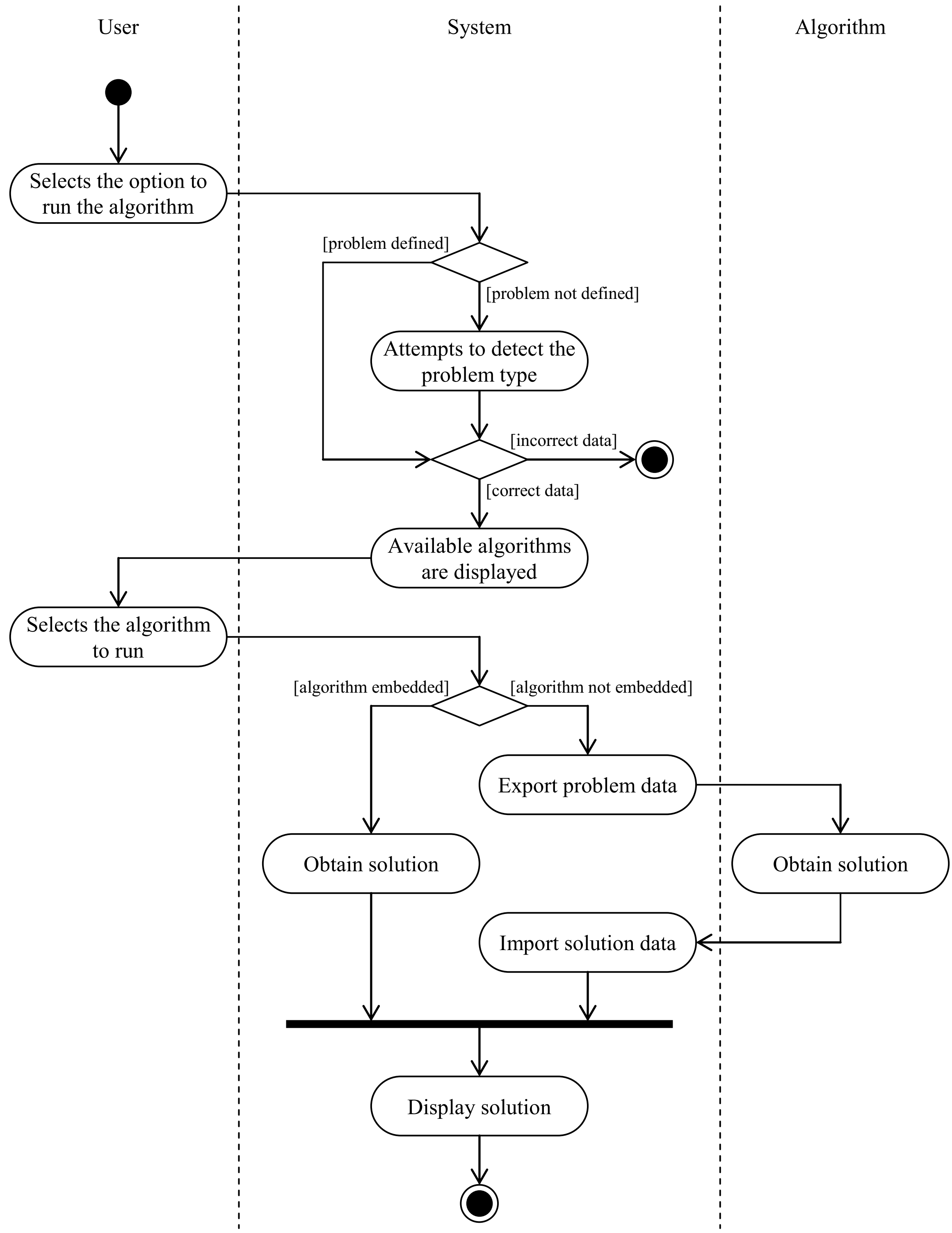


Figure 6.9 Activity diagram depicting how to obtain solutions in the DST.

During the development of the tool, the most used UML diagram was the activity diagram (as seen in Figure 6.9), which is a behavioural diagram that allows to graphically describe the workflows of stepwise activities. By showing the overall flow of the functionality (with a simple model) it facilitates the maintenance of more complex functionalities.

6.3.3 Iterations to Release

Most main development activities occur in this phase of the software development process, including modeling, programming, testing, and integration (system design and system implementation activities). These are performed in short iterations (one to three weeks long), with several of these iterations resulting in a release.

In the DST development, a high-level functionality was implemented at each iteration. At the end of each iteration, acceptance tests were performed with users. These tests were based upon the previously collected user stories. When users accepted the iteration, the functionality was integrated in the DST. Moreover, the next iteration would be prepared, with a new list of high-level functionality and corresponding priorities. This procedure allows the evaluation of the software to be performed throughout the development process.

Regarding the implementation (concerning programming), several coding standards and guidelines were defined. These were related to naming guidelines (for variables, objects, files, etc.), coding styles (code formatting, commenting, etc.), and language usage (variables, flow control, etc.). The goal was to enforce consistent style and formatting, thus helping to avoid common mistakes and improve code readability and maintenance, while not being overly restrictive. Moreover, code refactoring was commonly used to improve performance.

In this phase, preliminary testing in the form of unit tests occurred.

6.4 Summary

In order to effectively support decisions, a computation tool is required. In this chapter the development of such a tool was addressed. Firstly, the decision-making process was addressed, suggesting DSSs are more fit to be used when interaction between computers and DMs is required, as is the case when solving semi-structured problems. Since determining the location(-routing) of facilities is a semi-structured problem, it is therefore prone to be tackled using DSSs.

The correct development of DSSs was then addressed, namely their main components, involved stakeholders, and main development activities were identified. Several software development methodologies were analysed as well as their main advantages and disadvantages, allowing to draw conclusions on the most advisable method for the development of the intended tool.

HCI issues were also addressed, as the success of DSSs relies greatly on the ability to be easy to learn and use by target users. To that extent, several techniques that allow to test the usability of software were presented. In Chapter 7, one of these techniques (usability testing) is applied to test the usability of the developed tool.

Finally, the development of the DST was briefly described, where main development phases followed the XP methodology. Although not in strict compliance with the methodology, most of its defining practices were adopted in the development of the DST. Main development phases, namely, elaboration, planning, and iterations to release (and corresponding activities) were then described, setting guidelines for future developments of the tool.

In Chapter 7 the main components and functionalities of developed DST are described in detail, and results of usability testing presented and discussed.

Chapter 7

A Decision-Support Tool for Location-Routing Problems

In this chapter a decision-support tool (DST) for location-routing problems (LRPs) is presented. The tool aims at allowing the exploration of the solution-finding process in a way easily understandable by the user, enabling access to online geographic data through Web map services (WMSs). This DST was developed for Windows platforms having an architecture that easily allows the integration of new functionality.

The development of this tool intended not only to aid the solution-finding process but also to eventually foster greater insight of the problem at hand. This prompts the use of data/information visualization and human-computer interaction (HCI) methods. Moreover, by presenting approaches easy to work with and understand, the general public may have easier access and further understanding of the decision process involved in many of the current depot installation decisions.

The three main components of the proposed DST architecture are: the data structure, the models, and the user interface. The following sections will address in detail these main components. Firstly, the data structure and supported problems (based on the models) are discussed. Then, the graphical user interface (GUI), as well as its main functionalities, are presented. The GUI, developed to fit the target user's profile and intended tasks, is then evaluated with regards to its usability. The results of the usability testing concerning data input and visualization features are also examined.

7.1 Data Structure and Supported Problems

The developed tool has an open and modular architecture that allows to easily add or update new or existing functionality. The main components of the proposed DST's architecture can be seen in Figure 7.1, where interaction with other applications, input and output features, and algorithm integration are also depicted.

In the following subsections, the data structure and the supported problems (two of the DST's main components) will be presented.

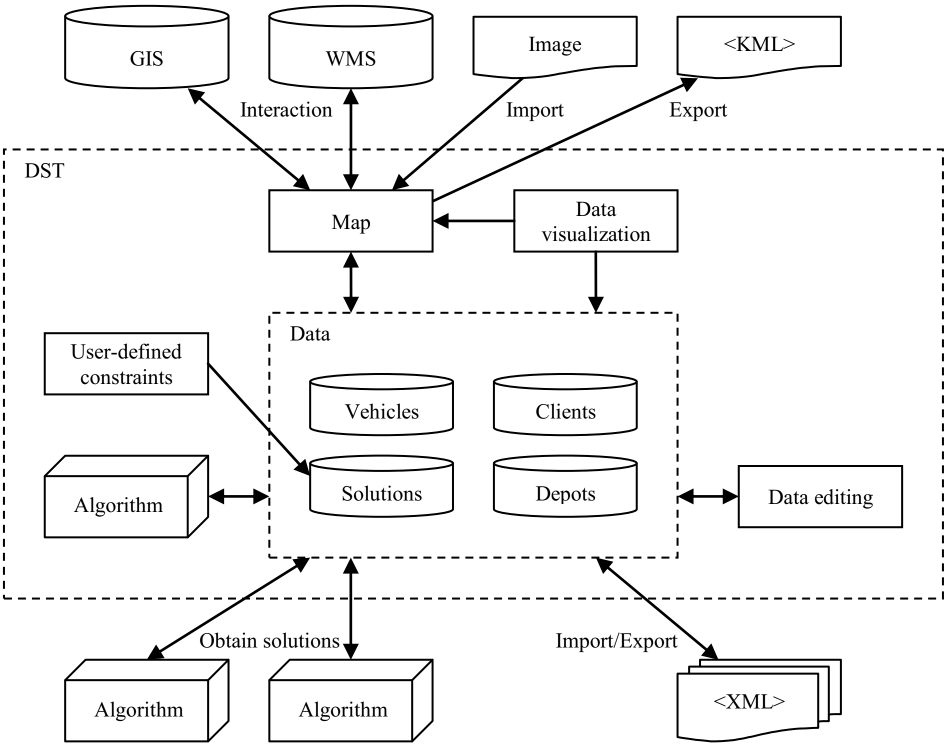


Figure 7.1 DST architecture (with its main components).

7.1.1 Data Structure

The tool presented in this chapter must support several data in order to solve instances of LRPs. Due to the complexity of the data to support, a data structure had to be defined having a set of characteristics: be flexible, in order to easily support future developments and file format evolutions, while maintaining compatibility; be able to hierarchically structure all the needed data, making it easier to interpret and maintain; and, be able to reflect its structure directly on the files obtained/generated by the tool, thus facilitating integration of new algorithms and interaction with other applications.

For these reasons, the data structure created for the tool is based on the extensible markup language (XML) file format. This choice is justified by the several advantages this data serialization format presents, namely:

- Freedom to define the data structure – allows to define any data structure without being restricted to a limited set of tags, which can later be validated by XML schemas.
- Robustness – makes easier to ignore and tolerate errors in the documents.
- Easiness of reading and editing – allows users to easily inspect files to change values or restore corrupted data (due to being human readable).
- Version interoperability – being extensible it becomes easy to add new functionality in the format without losing the ability to read files of previous versions.
- Operating systems interoperability – as XML is text-based it can be read in any operating system, even if the application that generated it is discontinued or incompatible.

- Transparent and documented content – allowing users/servers to easily inspect and verify its content.

By normalizing the data structure, future developments and interaction with other algorithms and applications is facilitated. Figure 7.2 depicts the data structure (and corresponding file structure) of the clients. The remaining main data/file structures can be found in Appendix D.

```
  <?xml version="1.0" ?>
  <!--  Created by Rui Borges Lopes (c) University of Aveiro  -->
- <Clients>
  - <Client>
      <Number>1</Number>
      <Name>Lisbon Transport Inc.</Name>
      <XCoord>1200</XCoord>
      <YCoord>2509</YCoord>
      <Colour>#FFFF8080</Colour>
    - <Demand>
        <Required>True</Required>
        <DistType>Uniform</DistType>
        <DistPar1>100</DistPar1>
        <DistPar2>0</DistPar2>
        <DistPar3>0</DistPar3>
        <DistPar4>0</DistPar4>
        <ServiceTime>10</ServiceTime>
        <SplitService>True</SplitService>
      </Demand>
    - <Supply>
        <Required>False</Required>
        <DistType>Deterministic</DistType>
        <DistPar1>0</DistPar1>
        <DistPar2>0</DistPar2>
        <DistPar3>0</DistPar3>
        <DistPar4>0</DistPar4>
        <ServiceTime>0</ServiceTime>
        <SplitService>False</SplitService>
      </Supply>
    </Client>
  </Clients>
  <!--  Clients:       Clients                                             -->
  <!--  Client:        Client                                              -->
  <!--  Number:        Client number                           int         -->
  <!--  Name:          Client name                             string      -->
  <!--  XCoord:        Client x coordinate                     double      -->
  <!--  YCoord:        Client y coordinate                     double      -->
  <!--  Colour:        Client colour                           string      -->
  <!--  Demand:        Client demand                                       -->
  <!--  Required:      Client requires demand service          boolean     -->
  <!--  DistType:      Client demand - distribution type       string      -->
  <!--  DistPar1:      Client demand - distribution parameter 1  double    -->
  <!--  DistPar2:      Client demand - distribution parameter 2  double    -->
  <!--  DistPar3:      Client demand - distribution parameter 3  double    -->
  <!--  DistPar4:      Client demand - distribution parameter 4  double    -->
  <!--  ServiceTime:   Time required to service the demand     double      -->
  <!--  SplitService:  Demand can be serviced more than once   boolean     -->
  <!--  Supply:        Client supply                                       -->
  <!--  Required:      Client requires supply service          boolean     -->
  <!--  DistType:      Client supply - distribution type       string      -->
  <!--  DistPar1:      Client supply - distribution parameter 1  double    -->
  <!--  DistPar2:      Client supply - distribution parameter 2  double    -->
  <!--  DistPar3:      Client supply - distribution parameter 3  double    -->
  <!--  DistPar4:      Client supply - distribution parameter 4  double    -->
  <!--  ServiceTime:   Time required to service the supply     double      -->
  <!--  SplitService:  Supply can be serviced more than once   boolean     -->
```

Figure 7.2 Data structure of the clients file.

The created formats are in strict compliance with the standards defined by the World Wide Web Consortium (2010).

7.1.2 Supported Problems

Although the proposed tool is mainly directed at supporting location-routing decisions, by doing so, it is also able to support closely related problems: the facility location problem and the vehicle routing problem (VRP). Regarding LRPs, the currently used data structure enables supporting (in squared brackets is the problem classification following the taxonomy proposed in Chapter 2):

- the round-trip location problem [1.1.1]
- the capacitated LRP (CLRP) [1.1.2]
- the location-arc routing problem [1.1.3]
- the plant-cycle location problem [1.1.6]
- the travelling salesman location problem [1.2.1]
- the stochastic LRP [1.2.2]
- the transportation-location problem [2.1]
- the many-to-many LRP [2.2]
- the multi-level LRP [2.4].

This set of supported problems can be, in some cases, easily adapted to fit other well known problems. For example, if the CLRP has only one possible depot location it becomes a capacitated VRP (CVRP), or the multi-depot VRP when several possible depot locations with no depot installation costs exist; likewise, if the CLRP considers only direct links between depots and clients, rather than routes, it is equivalent to the discrete location-allocation problem.

Thus, the data structure of the proposed tool can support location, routing, and integrated location-routing decisions (as they share the same elements, namely, clients, depots, and vehicles). Moreover, editing features of the GUI (which use the same input/visualization approach for all the problems) allow to maintain all the data required for these decisions. By addressing these decisions simultaneously it becomes possible to analyse logistics systems and determine which model(s) correspond(s) to a better approach for a specific scenario, with respect to location and routing activities.

The tool imbeds several algorithms for the aforementioned problems (e.g. the active guided search metaheuristic presented in Chapter 3); still in order to run algorithms, it is not required for them to be imbedded into the tool. By simply obeying to the data structure, algorithms need only to import and export the data files. Thus, the time required for the DST to obtain solutions is related to the algorithms' ability to solve the problem or the used commercial software (for integer programming).

7.2 Graphical User Interface

Upon developing this tool, the profile of the target user was taken into consideration. This user (typically a decision maker – DM) will be, in general, someone with higher education (not necessarily having a background on modelling and optimization), having a good knowledge of the problem at hand (or a large professional experience in real installations of depots and logistics systems design), at least reasonable computer literacy, knowledge of (Web) map applications, and which may use the tool infrequently.

According to this user profile, the information provided to users should neither be technical data regarding the used methods nor its validation; the focus should instead be on providing a usable interface, where the main usability goal should be easy and efficient access to solutions (results-oriented) and ease to learn and remember (Mayhew, 1992; Dix et al., 2004; Sharp et al., 2007). The profile of the target users, as well as the task they intend to perform using this DST, and usability principles (e.g. consistency, compatibility, familiarity, feedback, robustness, etc.) (Dix et al., 2004) were taken into consideration during the design of the tool.

The tool was developed for Windows platforms, using an extreme programming (XP) based methodology, and implemented in extensible application markup language (XAML), with C# as code-behind. Some of the concepts and features presented here were based on a DST intended to solve the CLRP (Lopes et al., 2008a).

The main objective behind the development of this tool is to allow to obtain, edit and visualize data, solve instances of the supported problems and visualize the corresponding results. To that end, the following main functionalities are provided:

- input (or edit) new (or existing) data in order to define the problem
- interact with WMSs, to obtain and visualize online geographical data
- obtain solutions and visualize them either through numeric or graphical representation
- visually compare different solutions
- allow user input to the solution obtaining process
- save (export) data to easily understandable (XML) files.

The conceptual model of the GUI is organized around a main window, with all functionalities accessible through the toolbar, or the menu (in a way easily understandable by users). This conceptual model, based on the information flow and similar map applications, aims to allow an easy and efficient access to solutions and it comprises four parts (Figure 7.3):

- a toolbar with buttons allowing a quick access to main functionalities (Figure 7.3, Area A)
- an extendable panel to edit/display data regarding the problems (Figure 7.3, Area B)
- a visualization area displaying information regarding the maps (Figure 7.3, Area C)
- a status bar with data regarding the used algorithm and objective function(s) value(s), if a solution was obtained (Figure 7.3, Area D).

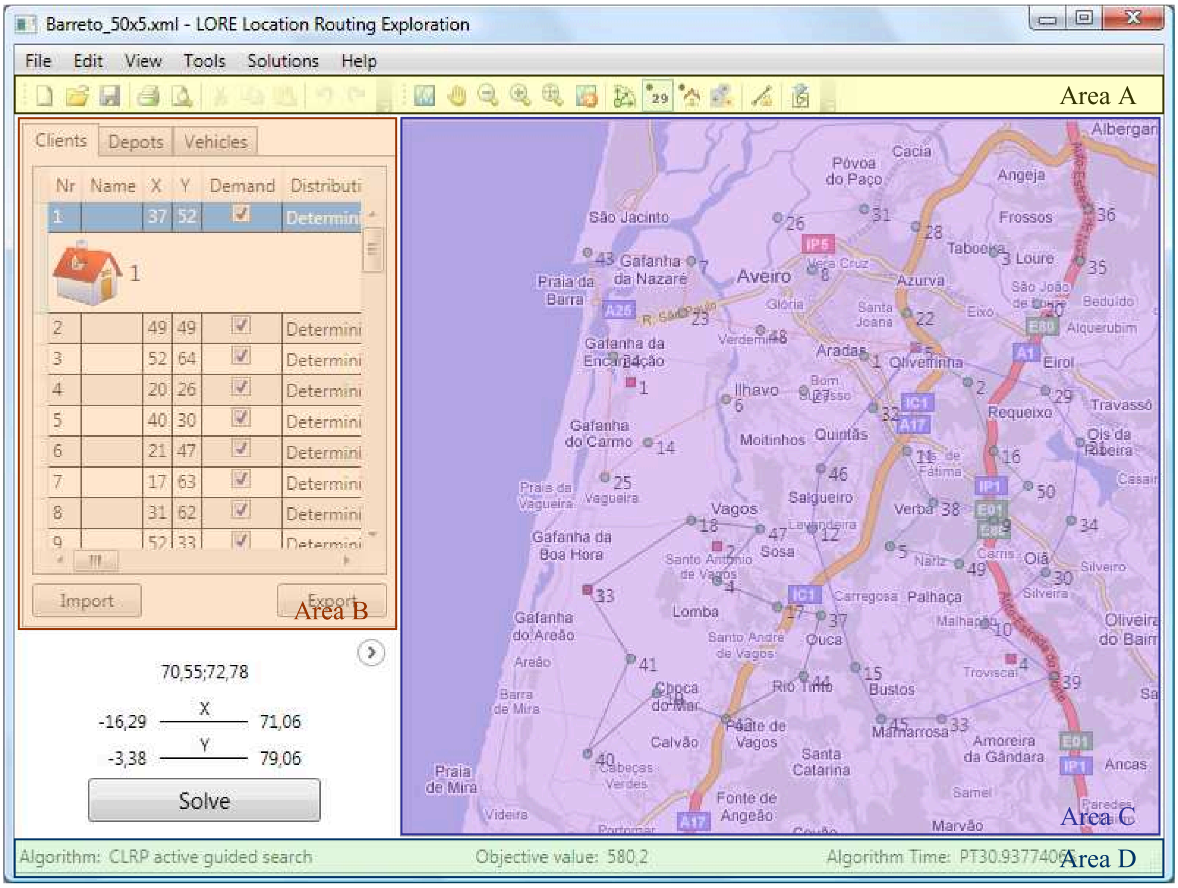


Figure 7.3 General aspect of the GUI and the four different main areas.

Regarding the toolbar (Figure 7.4), there is a set of buttons (with icons similar to commonly used applications, or having the same design concepts in mind) corresponding to different functionalities, besides the standard ones (New, Open, Save, Print, etc.).

Figure 7.4 Toolbar of the main window of the GUI.

These functionalities, which will be addressed as follows, are available from both the toolbar and the menu bar (thus providing greater flexibility and supporting users with different system experience and/or performing different tasks).

 **Import map:** allows to import an image to the map using standard formats (e.g. BMP, JPEG, PNG, or GIF).

 **Pan:** allows to pan the map (creates a panoramic effect of the map without changing the visualization scale).

/ **Zoom out/in:** allows to zoom out/in on a specific area of the map (i.e. changes the visualization scale). The interaction with WMSs provides new imagery and detailed data of the selected region.

Zoom to fit: allows changing the visualization scale in order to adjust the elements (clients, depots and vehicle routes) to the visualization area.

Hide map: allows to hide the map image in the visualization area, providing a better view of the different elements.

Objects size: allows changing the size of the graphical representation of the elements in the visualization area, thus facilitating their visualization.

/ **Display labels/images:** allows to activate or deactivate the visualization of labels/images corresponding to clients and depots;

View required service: allows viewing on the visualization area the service required by clients.

Fix arc in solution: allows to fix an arc between clients/depots which has to appear in the solution(s) to be obtained (i.e. allowing the user to define constraints).

Import solution: allows to import a solution file, possibly obtained with other software (provided the solution's data structure is obeyed).

The extendable panel allows to edit/display data regarding the elements in the problem (clients, depots and vehicles) using a data grid. The panel is most useful when inserting large quantities of data, where, by extending it, the user can work with (and visualize) a bigger data grid (Figure 7.5).

The visualization area, displaying information regarding the maps, will be addressed in detail later; while the status bar displays data regarding the used algorithm (name and time to obtain the solution), as well as the objective function(s) value(s) of the currently visualized solution.

Upon developing the GUI it was taken into consideration the guidelines defined by Microsoft for the development of Windows applications (Microsoft Corporation, 2010). Additionally, the GUI was firstly subjected to informal tests of adaptation to real target users; then, a formal usability evaluation was made (addressed in Section 7.3).

In order to obtain solutions, it is necessary to input all the data needed to specify and solve the problem. This step can be complemented with online geographic information obtained from interaction with WMSs.

After obtaining the data, several changes can be made from a visualization point of view (namely, associating graphical representations to the elements, changing the visualization scale, etc.), thus making easier the information interpretation. Solution(s) can be obtained (possibly incorporating information provided by the user), visualized, and compared with other solutions. Finally, the user can export all the edited/obtained data.

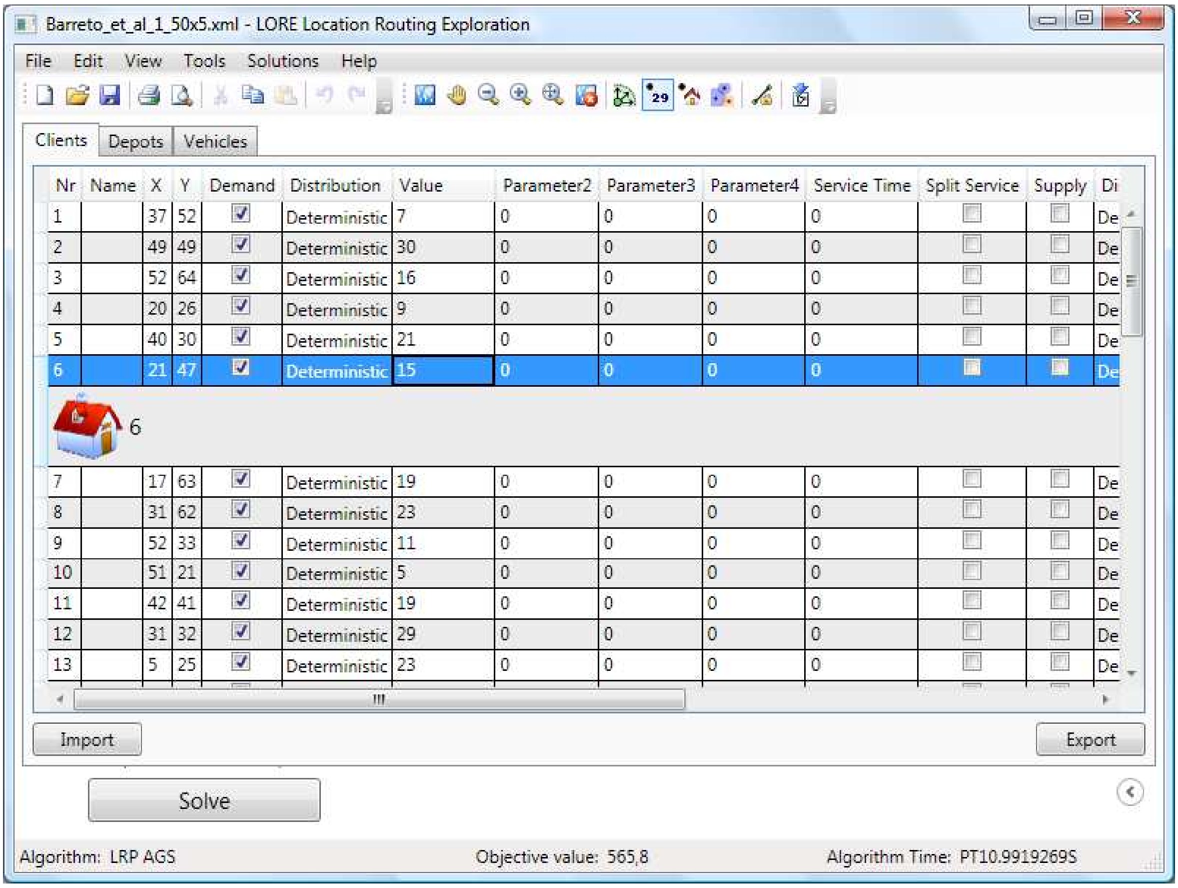


Figure 7.5 GUI with the left panel extended to edit/display data.

The following subsections will address, for each of these steps, the functionalities provided by the proposed DST, and how to use them.

7.2.1 Data Input

The supported data (which will allow defining the problem at hand) are the following:

- data regarding the clients to be serviced (coordinates, demand and/or supply distribution)
- data of the depots (coordinates, capacity and costs), already installed or to be determined, which will service the clients
- available vehicles and related data, namely, capacity, cost and where they operate
- distance matrix between clients and depots (by default Euclidean distances are assumed).

Obtaining (or maintaining) this information can be done using a data grid (more suitable to insert/edit large quantities of data), or directly on the map, where coordinates are directly obtained (or altered, by dragging clients and/or depots across the map) and a form appears to edit the remaining data. In all data input methods (seen in Figure 7.6) numerical feedback and visual representation are provided.

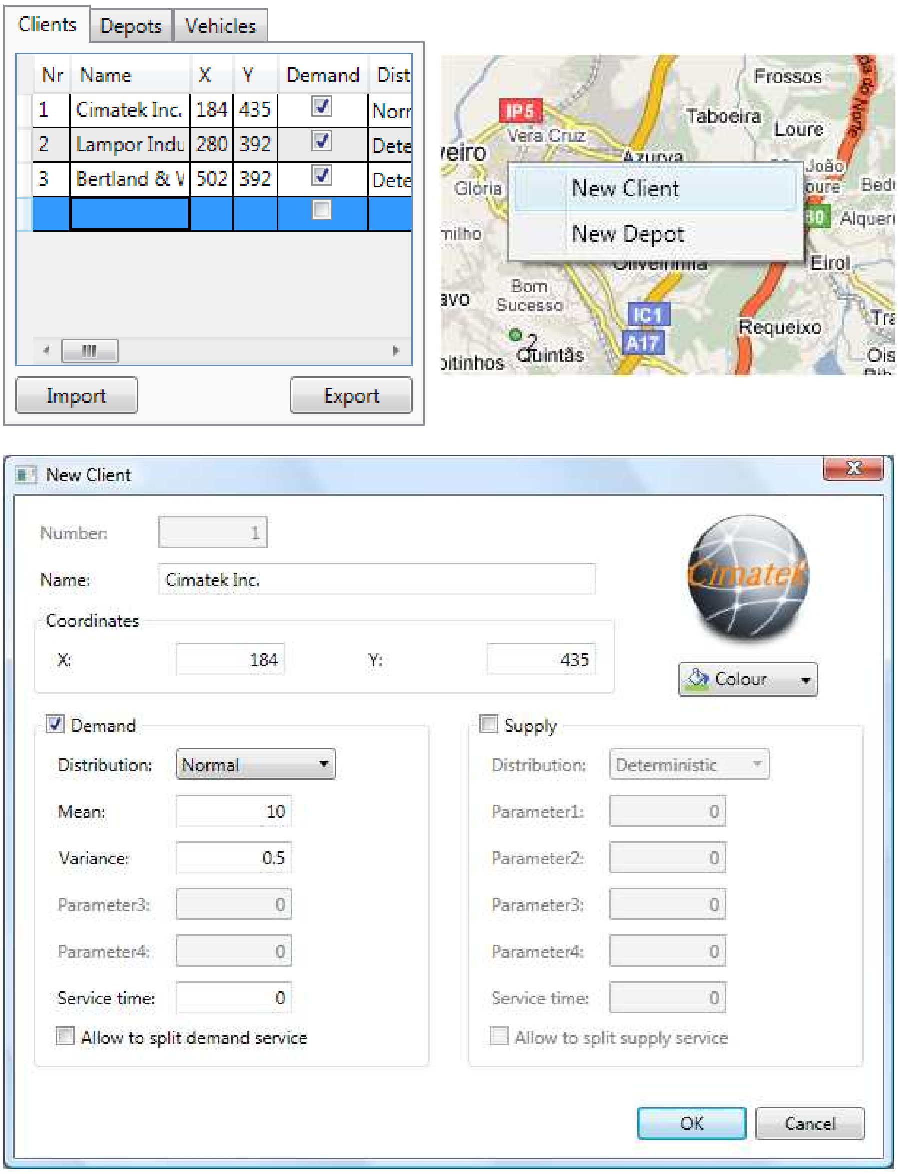


Figure 7.6 Data input using the data grid (top-left), the map (top-right), and the form (bottom).

In the last two cases (vehicle and distance matrix data), visual data input is not possible. It can be done however using data grid or specific forms (Figure 7.7), providing numerical feedback only, until a solution is obtained and (vehicle) routes are drawn.

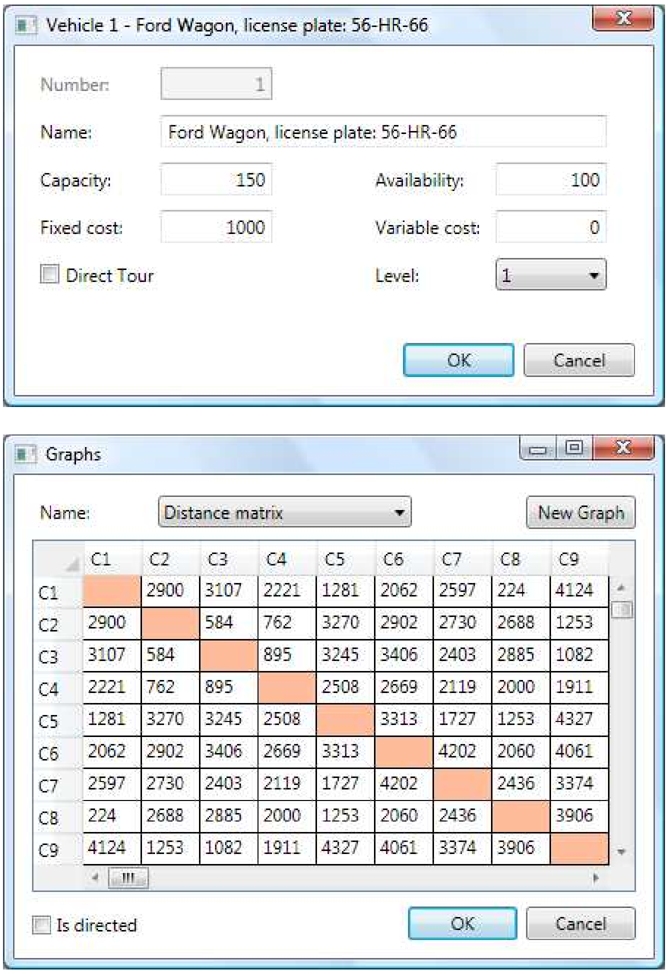


Figure 7.7 Vehicle (top) and distance matrix (bottom) forms.

7.2.2 Web Map Service Interaction

Geographical data are currently available online from many servers. When the servers follow the OpenGIS WMS standard (Open Geospatial Consortium, Inc., 2010), maps and requested information layers are made available in geo-referenced images (with standard formats: JPEG, PNG, etc.). The standard is based on a query syntax for posting a request for the desired layers and zoom window to the server, returning the corresponding map image. Taking advantage of interaction with WMSs, users can quickly obtain correct coordinates and distances regarding clients and depots. Moreover, additional information can be made available in layers (e.g. demography, road network, satellite imagery, etc.), even combining different online servers.

Figure 7.8 depicts the GUI interacting with the Demis Web Map Server (Demis, 2010), an OpenGIS WMS compliant.

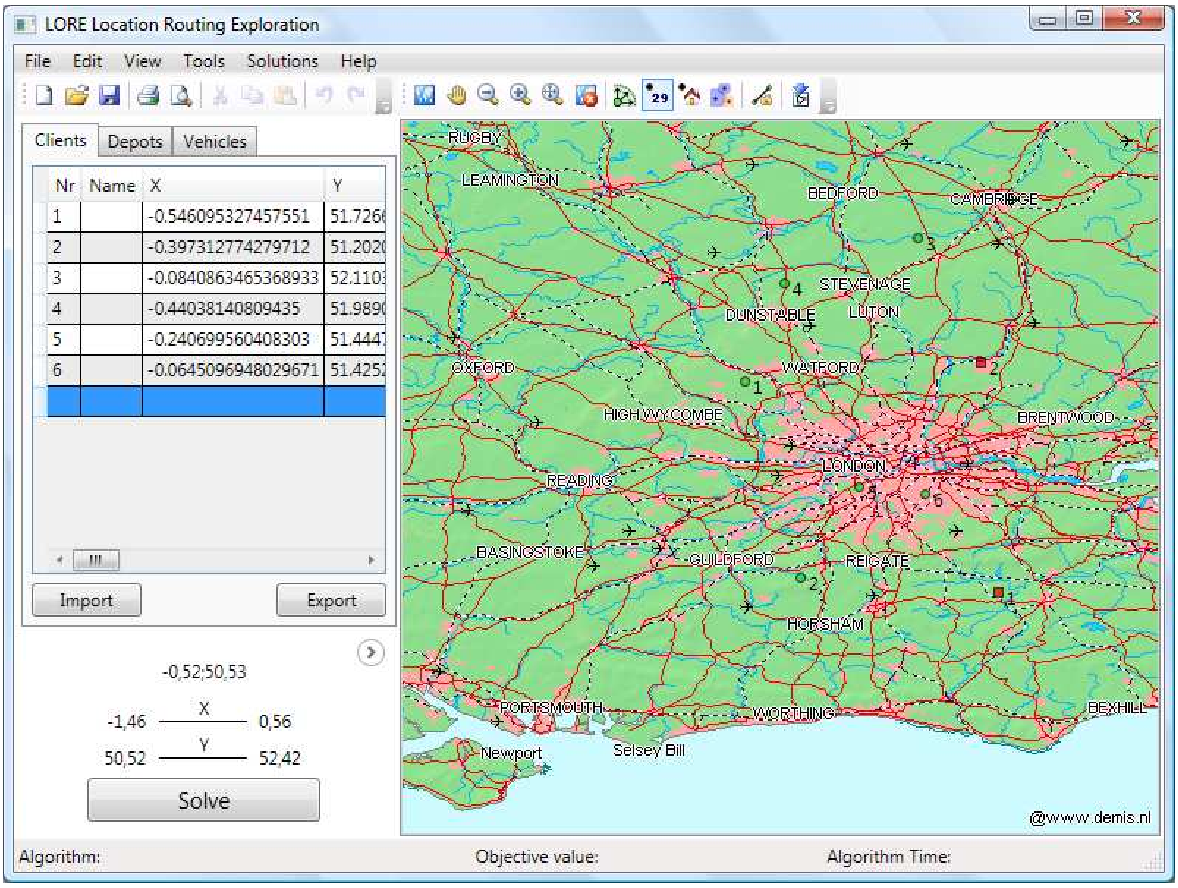


Figure 7.8 GUI interacting with the Demis Web Map Server.

By fully interacting with WMSs, the user can pan and zoom out/in any area of the world map and obtain the required imagery and geographical information.

7.2.3 Graphical Representation

Several representations for the elements involved in the process (clients, depots, and vehicle routes), as well as the map can be imported (Table 7.1). However, since superimposition of elements to the map may result confusing, transparency and size of their representation can be changed, allowing seeing the map underneath them; moreover, it is possible to define which are to be displayed simultaneously using a set of controls (presented previously). Users can also change the colour associated with the graphical representation of the elements, further distinguishing them from the map, as well as hide the map.

Table 7.1 Graphical representation of some elements in the visualization area.

Element	With label	With image	With label and image
Client			
Depot			

Another useful view is the service-based visualization which, for each client, displays a circle representing the required service, with the radius directly proportional to its value (Figure 7.9). Through this option users can easily visually identify clients with higher required service values.

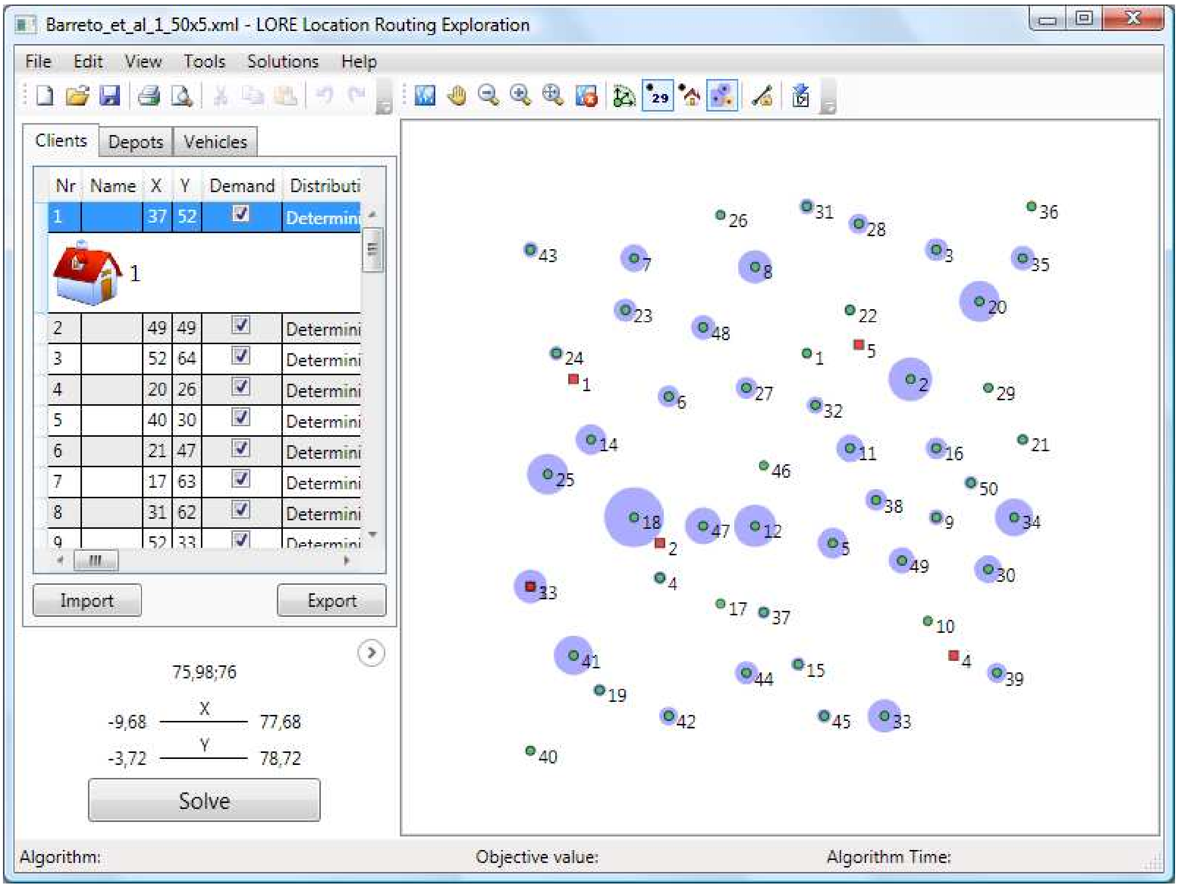


Figure 7.9 Visualization area displaying the service required by clients.

Finally, pan, zoom out/in, and zoom to fit allow fully evaluating the location of the elements in the map by changing the visualization scale.

7.2.4 Obtaining and Visualizing Solutions

Solution(s) can be obtained by:

- importing a solution file
- executing the algorithms.

Importing solutions from files makes possible to obtain the solution data from other (optimization) software packages, using the proposed tool to insert, edit or visualize the problem data. On the other hand, the possibility of directly executing one of the supported algorithms (where several runs can be performed, as randomness may be associated with the algorithms) enables the quick test of alternative scenarios.

As several types of problems are supported, it may be difficult for users to correctly identify the problem which they are addressing. For this reason, the tool abstracts the user (by default) from the choice of the type of problem (and consequently from the algorithms to use); still, the user is allowed to change it (top of Figure 7.10). After the type of problem (and corresponding model) has been obtained, users must define the desired objective(s) and algorithm (a list of algorithms able to solve the specific model is shown to users; as seen in the bottom of Figure 7.10).

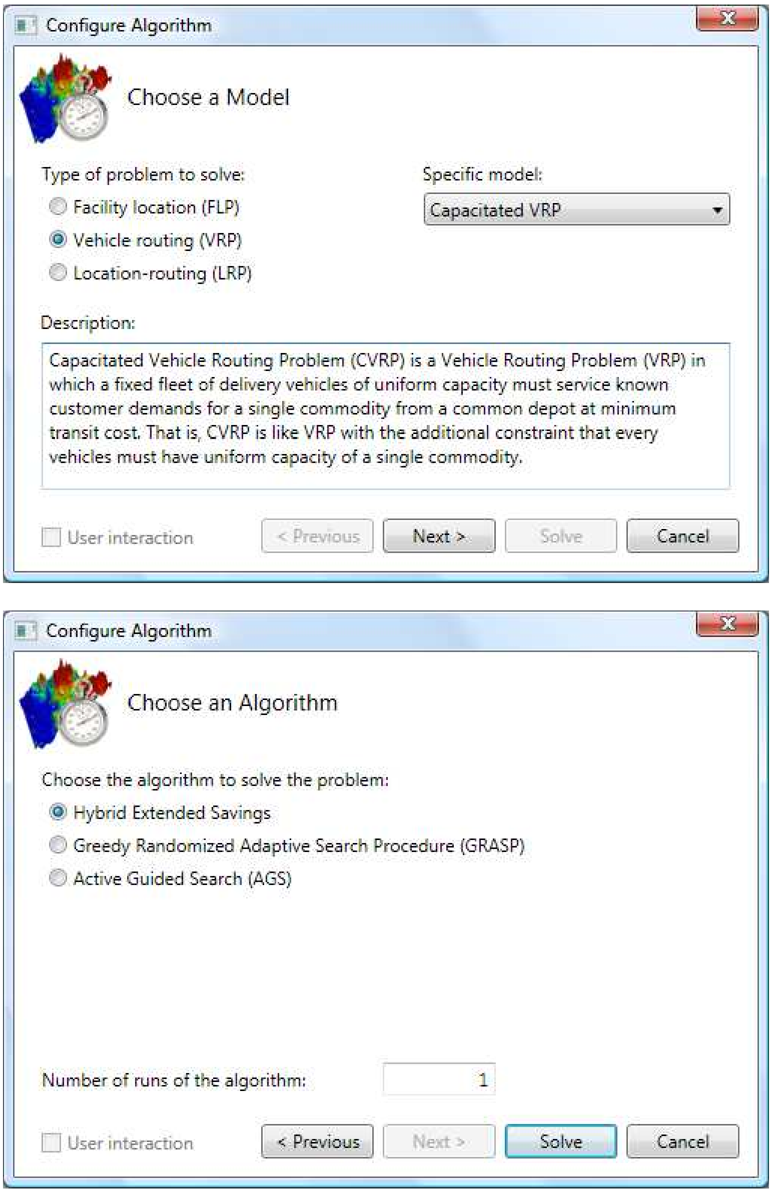
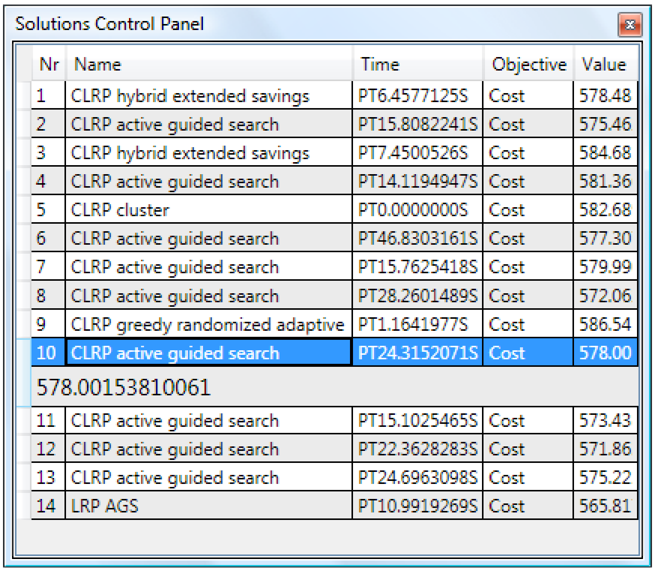


Figure 7.10 Choice and configuration of the type of problem (top) and algorithm (bottom).

After being obtained, an overview of the solution(s) data (algorithm name, time, objective function name and value) is displayed in a control panel (Figure 7.11). This allows the user to, at any given point, restore (or compare with) a solution previously obtained. Additionally, a solution data panel (Figure 7.12) displays all the data of the selected solution, namely, the total objective function value, depots to install and the vehicle routes data (capacity, assigned colour and tracing). Both options (control and solution data panels) use modeless dialog boxes (Microsoft Corporation, 2010) allowing a quick change between them and the main window. This type of window is most useful when performing frequent, repetitive and pending tasks (as obtaining a solution or comparing several ones, since a continuous analysis needs to be performed).

A solution can also be graphically visualized on the visualization area (Figure 7.13, left), where lines with different colours (for different routes, in order to facilitate interpretation of solutions) link clients and depots, representing the vehicle routes. Moreover, more than one solution can be simultaneously displayed, allowing visual comparison (Figure 7.13, right). This may also help to identify links common to all "good" solutions, that may be removed from the problem, reducing its size and difficulty.


Solutions Control Panel

Nr	Name	Time	Objective	Value
1	CLRP hybrid extended savings	PT6.4577125S	Cost	578.48
2	CLRP active guided search	PT15.8082241S	Cost	575.46
3	CLRP hybrid extended savings	PT7.4500526S	Cost	584.68
4	CLRP active guided search	PT14.1194947S	Cost	581.36
5	CLRP cluster	PT0.0000000S	Cost	582.68
6	CLRP active guided search	PT46.8303161S	Cost	577.30
7	CLRP active guided search	PT15.7625418S	Cost	579.99
8	CLRP active guided search	PT28.2601489S	Cost	572.06
9	CLRP greedy randomized adaptive	PT1.1641977S	Cost	586.54
10	CLRP active guided search	PT24.3152071S	Cost	578.00
578.00153810061				
11	CLRP active guided search	PT15.1025465S	Cost	573.43
12	CLRP active guided search	PT22.3628283S	Cost	571.86
13	CLRP active guided search	PT24.6963098S	Cost	575.22
14	LRP AGS	PT10.9919269S	Cost	565.81

Figure 7.11 Solutions control panel displaying currently obtained solutions.

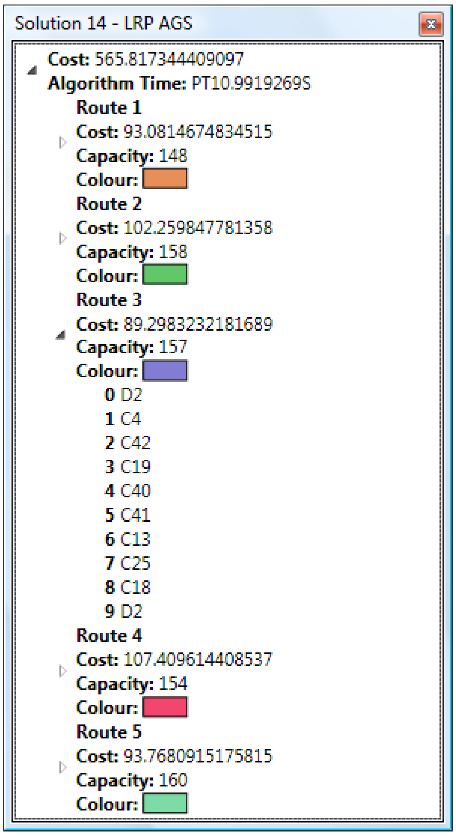


Figure 7.12 Solution data panel displaying (in tree view) all the data of the selected solution.

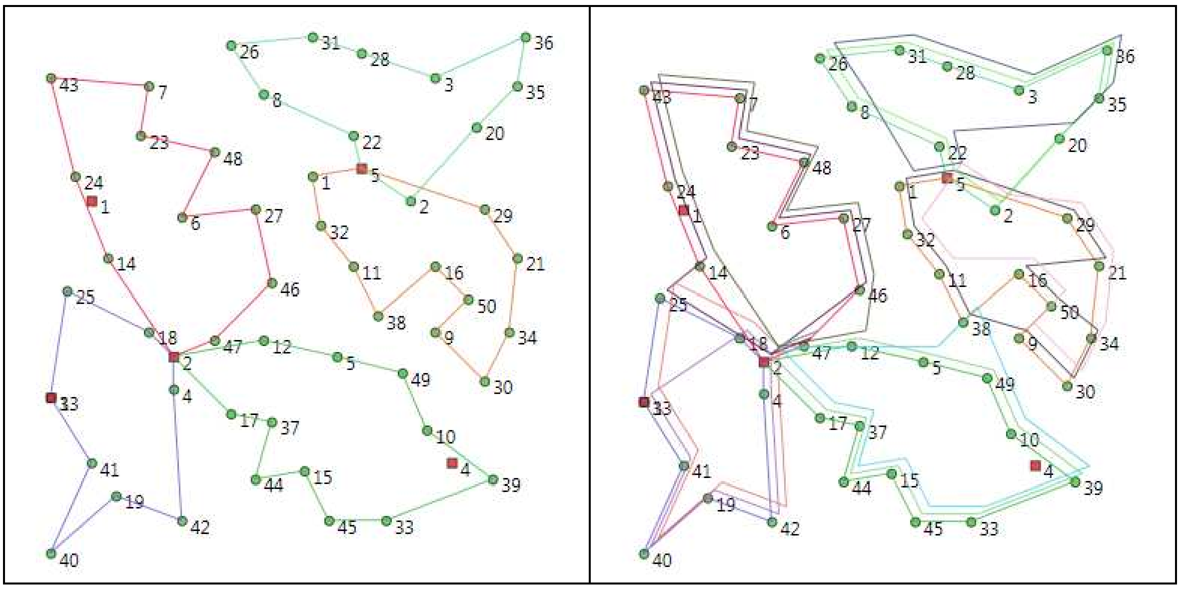


Figure 7.13 Solution graphical representation (left) and visually comparing three different solutions (right).

Finally, users can contribute to the process of obtaining the solution (possibly capitalizing on their experience in the case under study or reflecting constraints not considered in the model). In order to do so, they have to define the links that must appear in the final solution and, henceforth, all obtained solutions will include them (e.g. some clients are to be serviced in a predetermined order). Figure 7.14 depicts this procedure where clients 3 and 20 were forced to be serviced sequentially, and a solution was obtained obeying it.

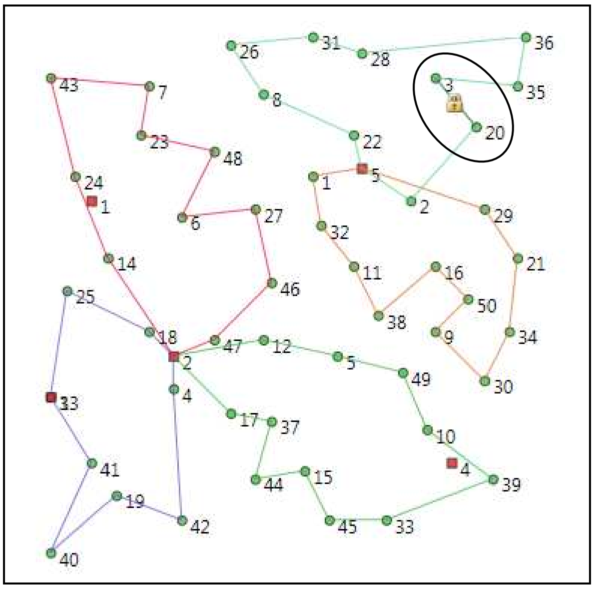


Figure 7.14 Solution where the user defined clients 3 and 20 are to be serviced sequentially (seen in the top-right corner).

7.2.5 Data Output and Other Characteristics

Regarding data output, the following options are available:

- printing the visualization area
- exporting data
- saving data to a file.

All the information displayed in the visualization area, namely, elements graphical representation, map and solution(s), can be printed; and data regarding clients, depots and vehicles can be exported to easily understandable XML format files (following the previously presented data structure) facilitating the integration with other applications.

Saving all the data to a single file (equally with XML format) is also possible. This allows users to easily recover previously saved problems, as well as editing the corresponding data outside of the tool.

Finally, besides the previously mentioned options, other characteristics have been added in order to improve the tool's usefulness (e.g. the ability to define a different language, the possibility of integrating other algorithms, and a help file documenting the tool).

7.3 Usability Evaluation

The GUI and all its visualizations were developed taking into consideration usability principles and guidelines, as well as feedback from several users. Usability testing was also performed in order to understand its potential and limitations. Initially, heuristic evaluation (Dix et al., 2004) was made by two evaluators with knowledge in usability (producing a list of possible usability problems), then, usability testing was made with the collaboration of a total of 50 users. Users were computer engineering students attending an introductory course on HCI. The choice was due to being users with high computer literacy, knowledge of HCI issues, and some experience with map applications, albeit possessing little knowledge of the problem at hand. This set of users may help find more easily usability problems, as well as test the ability of users with little or no experience on real-world depot installation decisions to use the tool.

In the usability testing, observation techniques and questionnaires (Ware, 2004; Mitchell, 2007) were used to evaluate data input, some of the adopted interface objects, ease of navigation, and some of the proposed visualizations. Two different sets of tasks were devised for users to perform. The first was composed of a set of 16 tasks, while the second had 20 tasks. The tasks were relatively simple, yet regarded representative of the most common operations to be performed by target users.

Users had to complete each task in a given time window, and were observed concerning the following data in each task:

- time required to perform the task
- whether completion of the task was successful
- if mistakes were made
- if the user felt lost
- if the user requested help from the observer
- difficulty to complete the task (both judged by the user and observed).

After completion of all tasks of the given set, users were asked about age, gender, and previous experience with map applications; and to evaluate several features using a qualitative scale:

1 2 3 4 5 N
○○○○○ ○

where "1" is complete disagreement, "5" is complete agreement and "N" corresponds to not having an opinion or not wanting to express it. Moreover, users were encouraged to provide comments or suggestions.

Results regarding users' opinion, as well as the list of tested features, can be found in Table 7.2, from where a general positive opinion can be inferred.

Table 7.2 Users' opinion on general and specific aspects of the GUI.

	Feature	Median
A	Is easy to learn	4
B	Organization is understandable	4
C	Response time is reasonable	3
D	Is easy to use	4
E	Is easy to insert large amount of data	3
F	Information layout is adequate	4
G	Help is needed using some functionalities	3
H	Further specific knowledge or tool usage experience is required	3
I	Text is easy to read	4
J	Amount of visible information is adequate	4
K	Icons used are easily understandable	4

Looking at Figure 7.15, depicting a cluster analysis dendogram for the users' opinion on general and specific aspects of the GUI, two groups can be seen (1 and 2). Group 1 encompasses aspects mainly external to the GUI (namely, user experience, WMS server responsiveness, and ability to insert large quantity of data), while group 2 concerns GUI-related features. Table 7.2 shows better median values for group 2, reflecting a general positive view of the GUI.

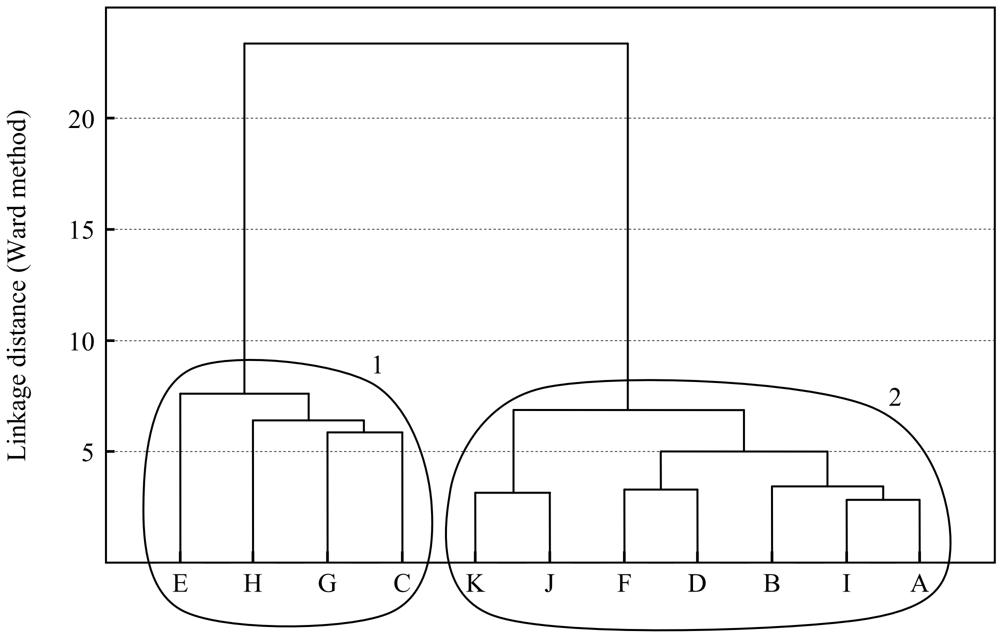


Figure 7.15 Cluster analysis dendogram for users' opinion on general and specific aspects of the GUI (Table 7.2).

Results on the two sets of tasks will be discussed in the following subsections, with the first set intended to test data input usability, and the second aiming at evaluating visualization aspects of the tool.

7.3.1 Data Input Evaluation

The first set of tasks mainly intended to test data input options. Eighteen users performed a total of 16 tasks. Some of the tasks are the following:

- loading and saving given files
- changing visibility of graphical representations of the elements
- inserting data regarding a single client or depot
- inserting/editing data regarding several clients and depots
- importing client, depot, and map graphical representations
- hiding and displaying information on the map
- identifying and deleting clients and depots.

The firstly mentioned two tasks were directed at evaluating some interface features (e.g. using menu and toolbar options, change viewing conditions); the following two tasks aimed at testing data input for both unitary and massive data insertion (thus testing data input using data grids and forms, as well as directly on the map); the last three intended to test ease of use of the interaction with the WMS and to identify information on the map.

The data obtained from user performance can be seen in Table 7.3 where, for each of the 16 tasks, is displayed: median required time (in seconds); number of users who completed correctly, with errors, felt lost or requested help; and median values for easiness felt and observed. Figure 7.16 shows the boxplots of the times taken by users in order to perform the different tasks. Observing Table 7.3, as well as Figure 7.16, it can be concluded that the most difficult tasks were related to the first contact with the different data input options (tasks 2 to 5)[2]. This may be due to the difficulty to understand some concepts regarding the addressed problem which led to some misunderstanding about the data to be inserted (mainly in task 4), which may not happen to users with experience on location decisions.

On the other hand, (task 6) very few users had difficulty completing the massive input of data (task 12) (although time to completion somewhat varies, mainly due to the users' different ability to quickly insert data using the keyboard), meaning the previous data input experience helped significantly.

[2] Task 6 was similar to tasks 4 and 5; however it may be concluded that users did learn from those previous tasks and therefore, data concerning task 6 do not show any difficulty.

Table 7.3 Data concerning user performance for the first set of tasks.

		Completed				Easiness	
Task	Time	Correctly	With errors	Felt lost	Req. help	Felt	Observed
1	9	17	0	3	0	5	5
2	36	17	1	2	0	5	5
3	15	16	1	5	0	4	5
4	69	10	5	6	2	4	4
5	42	14	4	3	0	5	4
6	34	18	0	0	0	5	5
7	6	18	0	0	0	5	5
8	40	17	1	2	0	5	5
9	10	17	1	2	0	5	5
10	57	15	3	2	1	5	4
11	56	15	3	3	0	5	4
12	120	16	2	4	1	5	5
13	31	18	0	2	0	5	5
14	41	17	1	2	0	5	5
15	8	18	0	1	0	5	5
16	23	14	3	6	1	5	4

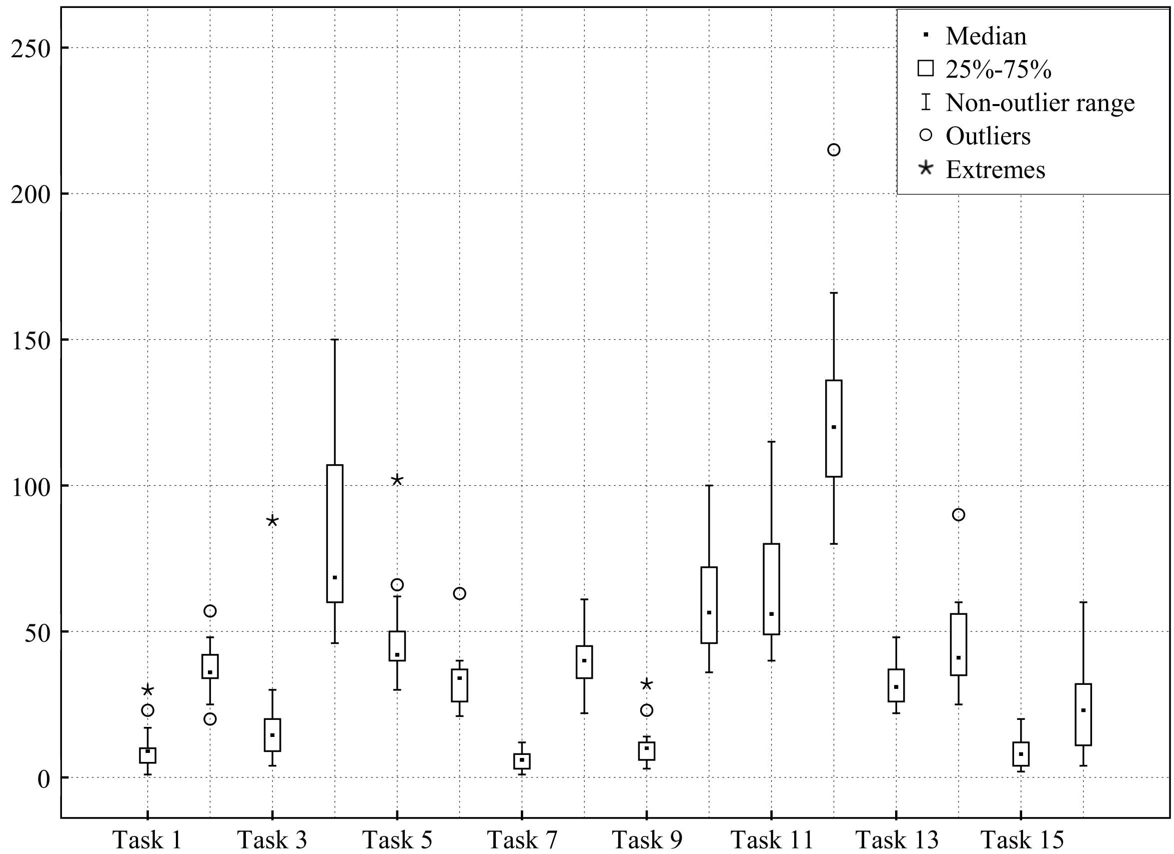


Figure 7.16 Boxplots of time spent, in seconds, in each task of the first set of tasks.

Accessing menu and toolbar options was considered trivial (tasks 1, 7, 8, and 9). Tasks 10 and 11 were related with interaction with the WMS where slow server responsiveness led to significant time variations. Tasks 13 to 15 aimed at testing the ability of identifying information on the map while task 16 regarded visibility options, which performed slightly worse due to the icons used to identify the functionality (found by some users to be inadequate).

Users' easiness felt values concerning the 16 tasks (Table 7.3) can be seen in Figure 7.17 where all mode values are 5, stating a general favourable perception of the GUI; task 4 has the least positive opinion and tasks 7 and 15 were considered the easiest. The null hypothesis of equal median of the 16 tasks is rejected by the Friedman test (non-parametric ANOVA) for a significance value (α) of 5%, as the p-value < 0.00004. Ordering the sum of ranks (Table 7.4) one can confirm that task 4 was felt as the most difficult one, followed by task 3. On the other hand, tasks 7 and 15 were considered the easiest.

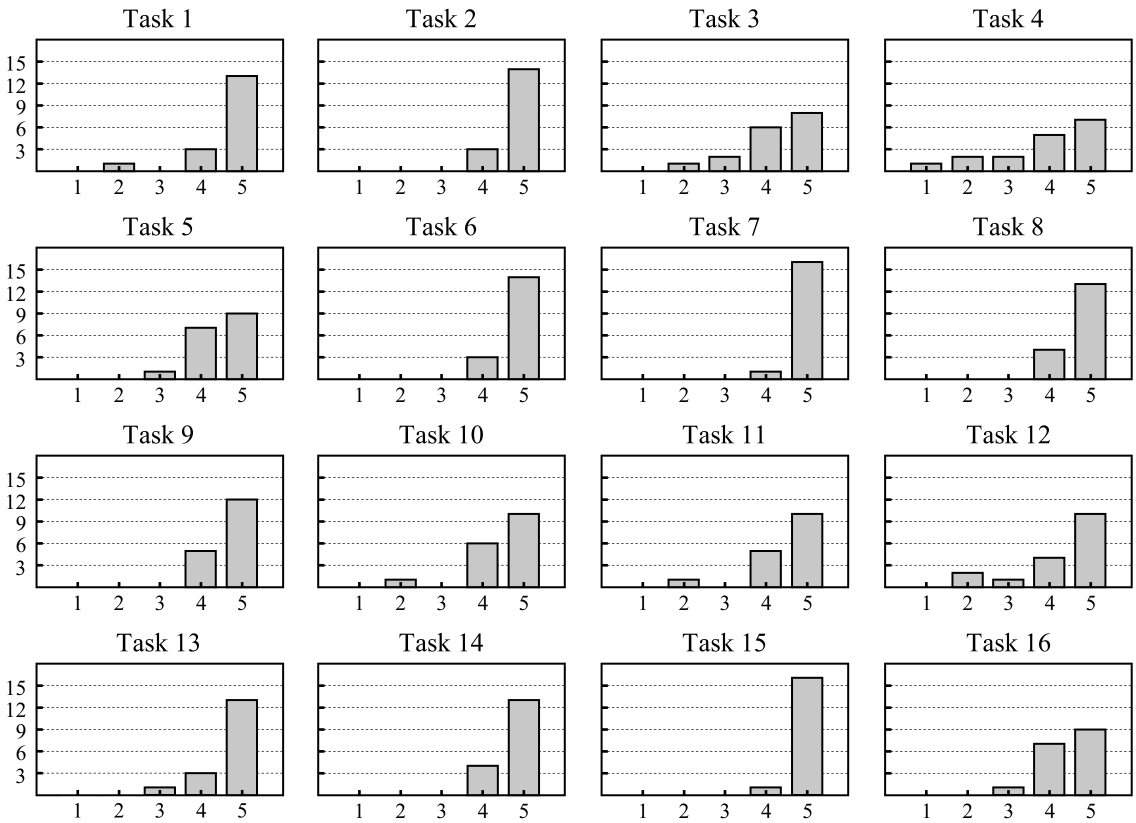


Figure 7.17 Bar charts of easiness felt values of the first set of tasks.

Table 7.4 Sum of ranks for the Friedman test concerning easiness felt values of the first set of tasks.

Task	Sum of ranks
4	89.5
3	102.5
5	116.5
10	117.5
16	122.0
12	122.5
11	123.5
9	139.0
13	146.5
8	147.5
1	149.0
14	149.0
2	154.5
6	156.5
7	170.0
15	170.0

Table 7.3 suggests a good agreement among the felt and observed easiness. This was statistically confirmed using the Wilcoxon matched pairs test for felt and observed values; the null hypothesis of equal median was not rejected for a significance value (α) of 5%, except for the pair "task 2 felt - task 2 observed" (with a p-value = 0.043)[3]. As can be shown in Figure 7.18 the perception of the users was more favourable (left bar chart) than the observers (right bar chart).

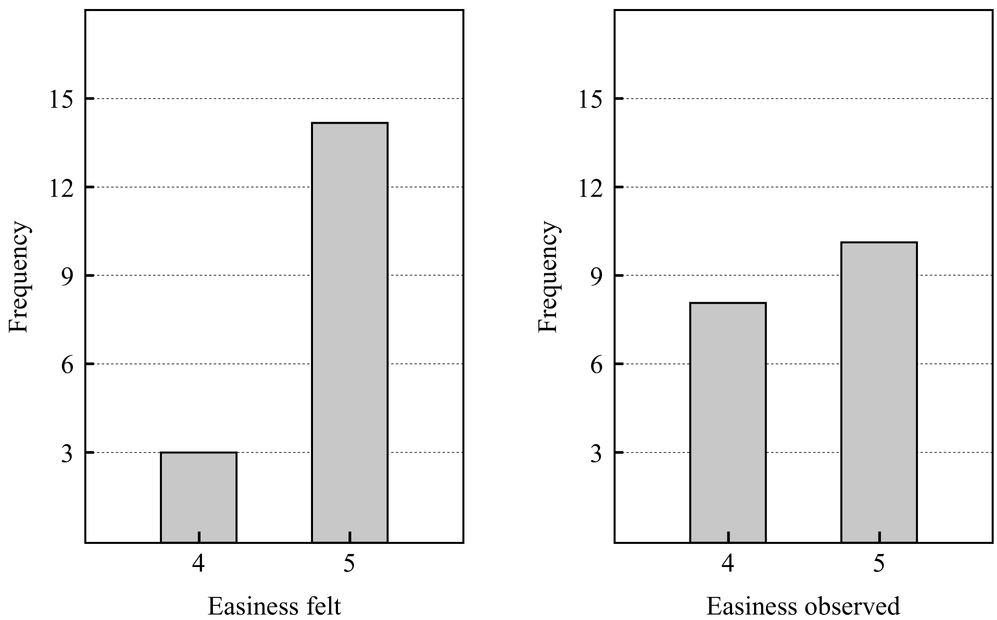


Figure 7.18 Bar charts of easiness felt (left) and observed (right) values for task 2 of the first set of tasks.

[3] This border value suggests that the sample should be increased; so, no definitive conclusions can be drawn.

Additional conclusions can also be drawn, namely, for massive input of clients or depots data, the preferred method was the data grid (used by around 88% of the users). To identify and delete clients and depots data, all users chose the data grid as opposed to the map (probably due to being easier and faster to identify). Still, the direct data input on the map was considered valuable since it makes easier to obtain coordinates of a specific location.

Looking at the obtained results, data input options can be regarded as adequate for the intended tasks and, consequently, for the operations to be performed in the DST.

7.3.2 Data Visualization Evaluation

In the second set of tasks, the main goal was to test the tool's ability to provide users with accurate data visualization features. The set was composed of 20 tasks which were performed by 32 users. Some of the tested tasks involved:

- interacting with the WMS (using zoom and navigation options)
- identifying clients and depots in the visualization area
- solving a specific problem using different algorithms
- obtaining information regarding the visualized solution (e.g. depots to install)
- obtaining information regarding a specific route
- changing the visualized solution (also to more than one simultaneously)
- changing graphical representation of the elements
- defining constraints on the problem
- obtaining information when elements overlap the map.

User ability to easily interact with the WMS was evaluated with the firstly mentioned task; while the following task regarded identifying information in the visualization area. Then, easiness of obtaining solution(s) was tested, and the following two tasks required information regarding (specific aspects of) solution(s) to be given by users. They were also requested to recover previously obtained solution(s) to the visualization area, changing graphical representations associated with the elements, and to define constraints on the problem. Finally, the ability to obtain information when elements overlap the map was tested.

Data regarding the corresponding user performance are given in Table 7.5, displaying, for each task, the same fields as Table 7.3. Boxplots with the time required by users to perform each task (of the second set) are shown in Figure 7.19.

Table 7.5 Data concerning user performance for the second set of tasks.

		Completed				Easiness	
Task	Time	Correctly	With errors	Felt lost	Req. help	Felt	Observed
1	10	31	0	5	0	5	5
2	150	5	19	25	13	3	3
3	113	24	7	10	1	4	4
4	30	26	6	8	4	5	5
5	54	15	14	26	14	3	3
6	15	29	2	4	3	5	5
7	40	30	2	0	0	5	5
8	11	29	1	5	2	5	5
9	21	26	6	9	4	4	4
10	60	23	8	19	6	3	4
11	166	8	24	29	15	3	3
12	30	15	10	16	7	5	4
13	20	22	9	10	10	4	4
14	57	19	10	12	6	4	4
15	16	27	4	5	2	5	5
16	64	12	16	16	4	4	3
17	20	27	3	6	1	5	5
18	10	31	0	3	1	5	5
19	18	25	5	5	2	5	5
20	23	23	7	3	1	4	4

Again, slow server responsiveness led to significant time variations in the interaction with the WMS (although selecting the feature, tested in the first task, was straightforward). Moreover, as the provided imagery, at some zoom levels, did not provide information on the location of important cities (required to perform tasks 2 and 3) many users felt lost. Task 5 results are due to the icon assigned to the feature (view required service), which most users had difficulty to find. Although the general view of solution data was easily accessible (tasks 8 and 9); regarding obtaining information on specific aspects of solutions, namely, cost, capacity and tracing of the routes (tasks 10 and 11), a high error rate and time variation occurred. This may be due to the need of using a different window to obtain the information, which users had difficulty to find.

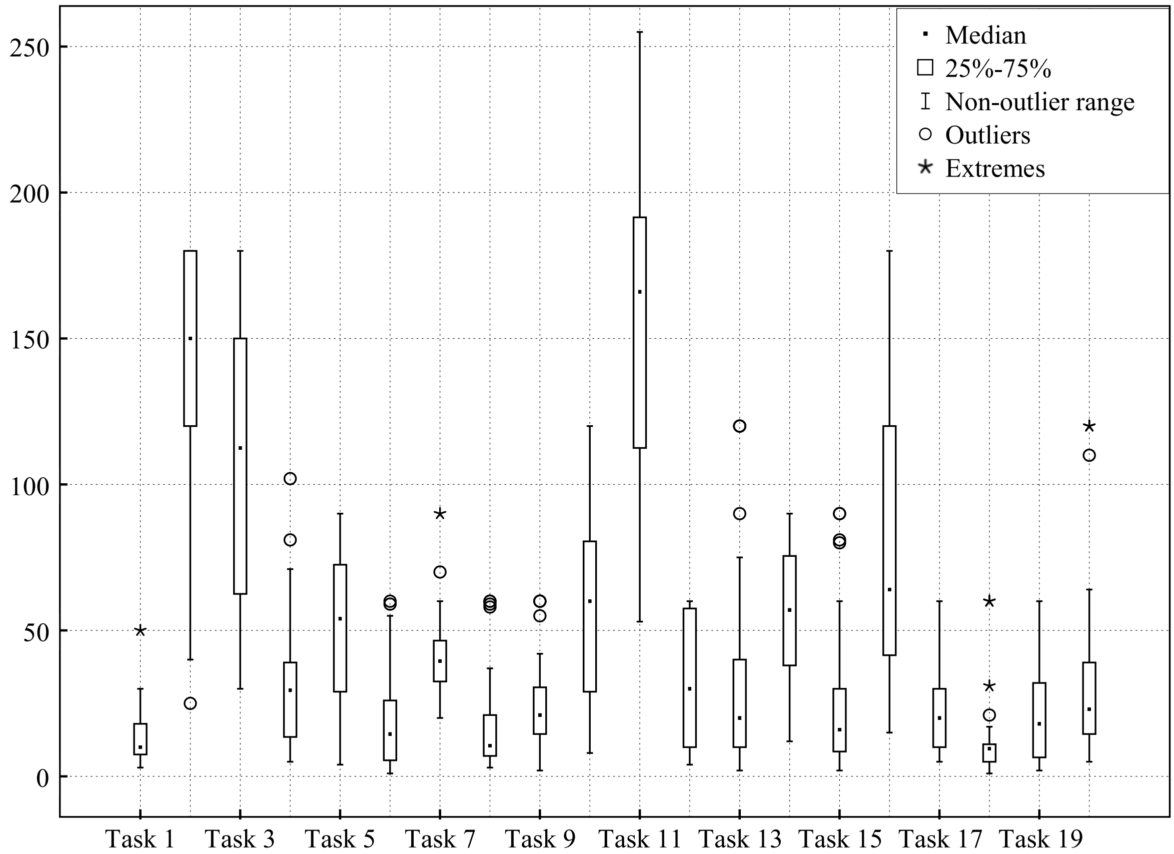


Figure 7.19 Boxplots of time spent, in seconds, in each task of the second set of tasks.

Recovering previously obtained solutions to the visualization area (tasks 12 and 13), which also required the use of a different window, was somewhat difficult. In task 16 users had to insert a constraint in the problem, and some difficulties were experienced, although they were able to easily identify it visually (task 17). Obtaining information from the visualization area was deemed easy (task 6), even when elements overlap the map (tasks 19 and 20); as was running the algorithms to obtain new solutions (task 7). Confirming results from the previous set of tasks, accessing menu and toolbar options was also considered trivial (tasks 4, 15, and 18).

User easiness felt values, concerning the 20 tasks (Table 7.5), can be seen in Figure 7.20 where 15 of the 20 tasks have a mode value 5, stating, once again, a general favourable perception of the GUI, being the easiest ones tasks 6, 7, and 18. On the other hand, tasks 2, 5, 10, and 11 were considered the most difficult ones. The null hypothesis of equal median of the 20 tasks is rejected by the Friedman test (non-parametric ANOVA) for a significance value (α) of 5%, as the p-value < 0.00000. Ordering the sum of ranks (Table 7.6), again tasks 11, 2, and 5 were felt as the most difficult ones. On the other hand, tasks 6, 18, and 7 were considered the easiest.

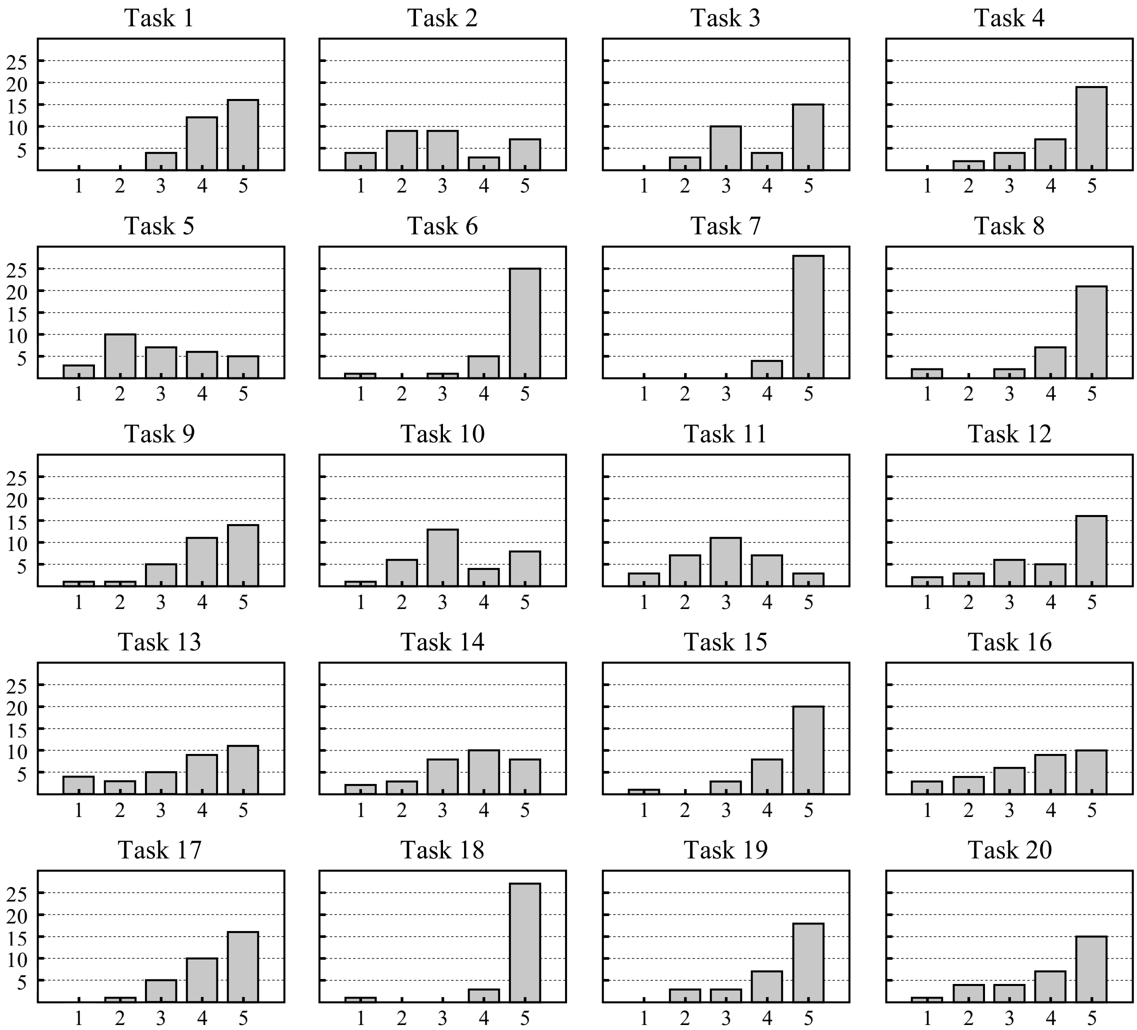


Figure 7.20 Bar charts of easiness felt values of the second set of tasks.

Once again, for this second set of tasks, Table 7.5 suggests a good agreement among the felt and observed easiness. This was statistically confirmed using the Wilcoxon matched pairs test for felt and observed values; the null hypothesis of equal median was not rejected for a significance value (α) of 5%, except for the pairs "task 1 felt - task 1 observed" with a p-value = 0.013 and "task 12 felt - task 12 observed" with a p-value = 0.020. For task 1, Figure 7.21, the perception of observers was more favourable (right bar chart) than the users (left bar chart). Concerning task 12, Figure 7.22, the perception of users was more favourable (left bar chart) than the observers (right bar chart).

Table 7.6 Sum of ranks for the Friedman test concerning easiness felt values of the second set of tasks.

Task	Sum of ranks
11	157.0
2	179.5
5	183.0
10	213.5
14	235.5
16	252.5
13	258.5
12	292.0
3	298.0
20	308.5
9	321.5
1	342.0
17	343.5
4	351.0
19	353.0
8	362.0
15	378.0
6	400.5
18	418.5
7	442.0

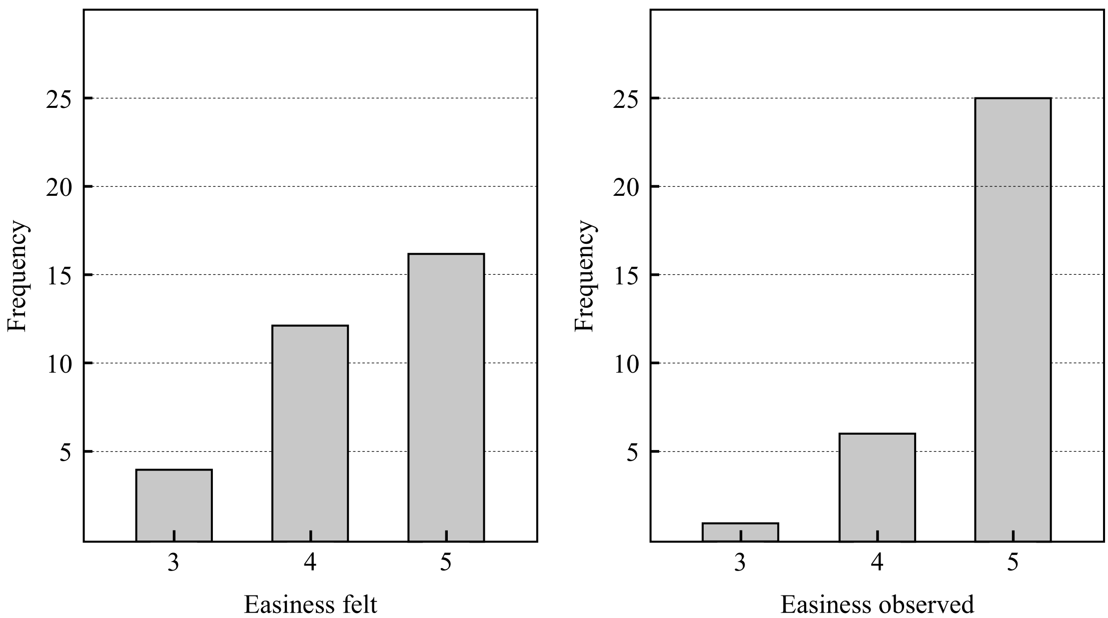


Figure 7.21 Bar charts of easiness felt (left) and observed (right) values for task 1 of the second set of tasks.

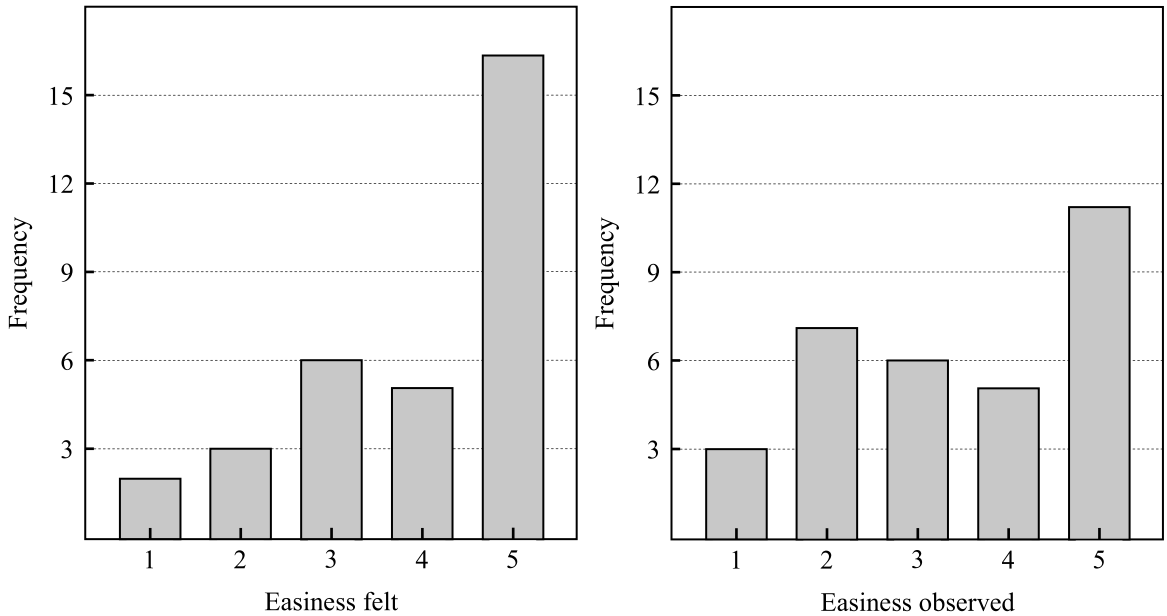


Figure 7.22 Bar charts of easiness felt (left) and observed (right) values for task 12 of the second set of tasks.

Results obtained in this set of tasks may lead to conclude that data visualization features are not as easy to use as data input options. This was mostly true when users attempted to obtain specific numerical results regarding the visualized solution. Also, it should be noted that, the users of the second set of tasks did not experienced inputting data (due to happen when using the tool in a real scenario), which may have somewhat hampered the results.

7.4 Summary

In this chapter a DST developed to address location-routing decisions is presented (accessible through the site: http://lore.web.ua.pt/). The tool was developed in XAML, with C# as code-behind (based on some XP methodology concepts and practices), and has three main components which are discussed here: the data structure, the models (general enough to supported several problems), and the GUI.

The data structure is based on XML and allows supporting several other logistics problems besides the most common LRPs (e.g. the CVRP, the multi-depot VRP, and the discrete location-allocation problem). Regarding the GUI, it allows the exploration of the solution-finding process in a way easily understandable by the user. Furthermore, the tool enables access to online geographic data through WMSs. A usable user interface was a great concern throughout the development of this tool, which was designed to allow decision-makers with a moderate computer literacy to be able to obtain good quality solutions without much learning effort. The profile of the target users (who may not have specific knowledge on the used methods) as well as the tasks they have to perform were taken into consideration in the development of visualization and interaction solutions.

Usability evaluation was performed which allowed finding usability problems and gather new ideas, which helped improve the proposal. Test results allowed identifying usability problems, obtaining new ideas, and, at the same time, attesting the ease of use and learnability of the adopted visualization solutions. Results were generally positive in the three dimensions of usability: ease of learning, ease of use, and satisfaction.

The main identified problem was concerning the use of additional windows to obtain specific numerical data. Firstly, by not having a toolbar button to those functionalities (only accessible from the menu), users had some difficulty in finding the corresponding feature. Secondly, as the users were starting to get used to having all the information in the main window, the use of other windows (even tough of a different type: a modeless dialog box) felt somewhat cumbersome. For future releases of the tool, highlighting selected clients and/or depots was also considered an interesting feature as it would help identify information on the map (due to the superimposition of these elements to the map).

The main limitation of the usability evaluation is that, although users had high computer literacy, and some experience with map applications, they had little knowledge of the problem at hand, and thus only partially fitting the target user profile.

As the best decision may not be the most cost efficient, this type of tool can help DMs make a more scientifically supported decision, by providing the total estimated costs of a set of different solutions (thus improving the DM’s insight and judgement). From this point it is up to the manager to make the decision, taking into consideration the estimated cost, the service level or even the company strategy and motivation.

Chapter 8

Conclusions

The location of facilities and the distribution of materials have been studied in this thesis. These are two highly related logistics activities that have been addressed using an integrated approach: the location-routing problem (LRP). With the purpose of supporting location(-routing) decisions of semi-obnoxious facilities, several approaches for LRPs were developed, and incorporated in a decision-support tool (DST).

In this final chapter, main conclusions regarding the developed approaches and decision support are summarized. During the course of the work presented in this thesis some limitations, future work, and promising research directions were also identified. These will be presented in the following subsections.

8.1 General Conclusions

The LRP has been attracting increasing attention from the research community (see Appendix A for a brief analysis). This problem has been studied using different objectives and constraints, representing different real-world scenarios, which has led to a somewhat disperse body of knowledge. With the taxonomy presented in this thesis, it becomes possible to segment the existing research, where several variants of the LRP have emerged as poorly studied, and foster future studies. Although mainly focused on the problems intrinsic characteristics, the taxonomy also categorizes papers according to the adopted algorithmic approaches and used objectives, which were analysed, and some conclusions drawn.

Regarding the different algorithmic approaches a separation was made between exact and heuristic, with different found methods being listed. As the LRP is NP-hard, exact approaches are restricted to small instances. In order to tackle bigger instances, heuristic approaches are required to be employed. However, due to the lack of comprehensive studies benchmarking solution quality, no absolute conclusion can be made on the effectiveness of a single method or approach. Moreover, the lack of strong lower bounds (for almost all variants of the LRP) makes increasingly difficult to conclude on the performance of heuristics for large instances.

Looking at objectives found in LRP papers, a natural separation is regarding the number of addressed objectives (single- or multi-objective), which leads to different models and approaches. This was also the separation adopted in the proposed taxonomy. The majority of single-objective models focused on cost (or a surrogate such as distance) minimization. Multi-objective models have revealed to be scarcely studied, with objectives falling into the category of cost minimization, environmental aspects or equitable distribution.

From the newly proposed LRP taxonomy two basic single-objective problems were chosen to be studied: the capacitated LRP (CLRP) and the location-arc routing problem (LARP). For solving the CLRP a new metaheuristic (active guided search) was developed, with results suggesting that the approach is competitive (providing best average results for two out of the three sets of benchmark instances, with reasonable computing times; new best results were also found; and several best known results matched). In order to solve the LARP, several constructive methods, heuristic improvements, and metaheuristic approaches were proposed. A new set of benchmark instances was also developed, which allowed to compare the different proposed approaches. The approach that performed best was the tabu search combined with a greedy randomized adaptive search procedure (outperforming the remaining, results wise, and being extremely competitive regarding computing times). The newly developed instances also appeared to be balanced (concerning location and routing costs) and representative of several different cost configurations.

The previous basic single-objective LRPs addressed only cost minimization. However, when considering the location of a semi-obnoxious facility, several other objectives gain relevance. By reviewing the obnoxious and semi-obnoxious facility location (and routing) literature, two other important objectives (besides cost) were identified: obnoxious effect minimization and equitable distribution (through the minimization of the maximum individual obnoxious effect). This suggests the use of multi-objective approaches in the location(-routing) of these facilities (being defined and modelled as a multi-objective CLRP).

Two types of approaches can be thought of when solving multi-objective problems (each corresponding to a different decision-making scenario): non-interactive and interactive methods. Approaches for both methods are proposed in this thesis.

The defined multi-objective CLRP was firstly solved using an evolutionary algorithm (a non-interactive method). Regarding the three addressed objectives, it could be concluded that obnoxious effect and equity seem positively correlated, while cost is negatively correlated with the remaining two objectives. Results also suggest that the study of the location of semi-obnoxious facilities should simultaneously consider routing, as the routes (and corresponding cost) may change significantly when different possible facility locations are considered.

Concerning interactive methods for multi-objective mixed integer programming (MOMIP) problems (as is the defined multi-objective CLRP), current state of the art was reviewed. The review pointed to an open communication protocol paradigm as the most appropriate for current decision-making scenarios. A new interactive multi-objective method, following the open communication protocol, was thus proposed. The newly proposed method relies on graphical and numerical information, and is not too demanding from the decision maker (DM) point of view. Computation wise, it uses a weighted sum program that enables to obtain all non-dominated solutions of any given MOMIP problem (being applied to a multi-objective CLRP test instance with a step-by-step example).

LRP approaches have been somewhat restricted to the academic community. However, the underlying location and routing decisions are essential for the correct management of organizations. In order to effectively support these decisions, a computation tool must be made available to DMs (from which the general public can also profit). Such a tool should be able to incorporate advanced algorithms and, at the same time, be easy to learn and work with. For the correct development of such a tool, firstly, the decision-making process was studied, where it could be concluded that a decision support system (DSS) is most applicable to this type of problems. Then, the development of a DSS application (DST) was analysed, with main DSS components, stakeholders, development activities, and software development methodologies being identified. Out of the main software development methodologies, extreme programming was found to be the most appropriate for the development of the intended DST (mainly due to the constant user feedback, reduced development time, and frequent releases).

Finally, the developed DST is presented, with its main components (data, models, and user interface) and functionalities discussed. The adopted data structure allows supporting not only (several variants of) the LRP but also other closely related problems. Regarding the graphical user interface (GUI), it allows the exploration of the solution-finding process in a way easily understandable by the user. This was one of the main concerns throughout the development of the tool, which was designed to allow DMs with a moderate computer literacy to be able to obtain good quality solutions without much learning effort. The profile of the target users (who may not have specific knowledge on the used methods), as well as the tasks they intend to perform, were taken into consideration in the development of visualization and interaction solutions.

To test the adopted visualization and interaction solutions, usability evaluation (usability testing) was performed, allowing to find usability problems and gather new ideas to help improve the proposal. Two sets of tasks were devised to test data input and visualization features, where data visualization features appeared to be not as easy to use as data input options. This was mostly true when users attempted to obtain specific numerical results regarding the visualized solution. Nevertheless, overall results attest the ease of use and learnability of the GUI, being generally positive in the three dimensions of usability: ease of learning, ease of use, and satisfaction.

8.2 Limitations

During the development of the work presented in this thesis some limitations were identified, which will be enumerated, as follows, for future consideration.

The metaheuristic presented to tackle the single-objective CLRP, although appearing competitive when comparing with other approaches in the literature, may require further testing, mainly on real-world instances (as existing sets of benchmark instances are mostly randomly generated). This potential limitation derives from the fact that tuning of the metaheuristic took into consideration results obtained in the specific sets of benchmark instances, which may have led to biased results. This may, however, also hold true for the remaining approaches in the literature.

Regarding the LARP, conclusions on the proposed approaches were based on the results obtained for the newly devised set of benchmark instances. However, not only the instances lack lower bounds, but also the developed upper bounds used the same local search methods (although within different frameworks). This may prevent from allowing to draw absolute conclusions on the performance of the proposed approaches, as well as the applicability of the instances.

The benchmark instances used to evaluate the proposed multi-objective CLRP evolutionary algorithm, were devised envisioning the location of desirable facilities. Therefore, for undesirable (obnoxious and semi-obnoxious) facility location (the underlying decision of the defined multi-objective CLRP), although the clients structure might hold, the possible depot locations will eventually be different. Moreover, lacking results from other multi-objective approaches, the proposed algorithm could only be compared with regards to one of the objective functions (cost) and the number of generated non-dominated solutions.

Concerning the interactive multi-objective method (presented for tackling MOMIP problems), although, from a numerical point of view, being equally valid for more than three objectives, the graphical representation may result confusing. Moreover, the number of subregions to be eliminated may be less relevant as the total number of obtained subregions greatly increases with the number of objectives considered, while still only two of the subregions are to be eliminated by unfeasibility and dominance. Another limitation of the proposed method is that the three objective proposal may require dedicated graphical tools, as by hand drawing of the three-dimensional object may be somewhat confusing and error prone.

Finally, regarding the developed DST, usability evaluation was performed (using two sets of tasks) in order to validate the GUI and obtain new ideas. Obtained results were encouraging on both sets of tasks, although performing worst in the second set. Two limitations of the study were: the users of the second set of tasks did not experienced inputting data, which is due to happen when using the tool in a real scenario; users had high computer literacy, and some experience with map applications, yet little knowledge of the problem at hand, and thus only partially fitting the target user profile.

8.3 Future Work

Encountered limitations and ideas led to consider some future work and identify promising research directions, which will be addressed in this last section.

Development of new sets of benchmark instances may prove useful, as the existing sets are mostly randomly generated. New sets should either be based on real-world problems, or generated using a structure that allows to draw conclusions on which real-world situation is more fit to be tackled with a specific approach. In the latter case, in order to allow to make inferences, a sufficiently large sample of instances with different cost structures, spreading and grouping patterns is required. For the different LRP variants addressed in this thesis, this need was identified for the (single- and multi-objective) CLRP and the LARP.

The development of new sets of benchmark instances would allow to further test the proposed approaches, as well as draw more conclusions on their applicability to specific scenarios.

New lower and upper bounds to tackle the LARP and the defined multi-objective CLRP should also be developed. For the LARP, it would enable to further test the proposed instances and fine-tune the approaches. Regarding the defined multi-objective CLRP, a deeper analysis could be provided, as the existence of other approaches would allow to compare results using most common quality metrics in the multi-objective literature, namely, distance to the Pareto-optimal front and diversity (space covered of the obtained non-dominated front).

The evolutionary algorithm, proposed for tackling the defined multi-objective CLRP, could be improved with the inclusion of local search. It has been proven, in the single-objective literature, that genetic algorithms perform better when hybridized with local search procedures (although hybridization is still rarely used in multi-objective optimization).

Regarding the interactive multi-objective method, and following the work by Ferreira (1997), a new proposal can also be made. By applying the Tchebycheff metric, instead of the weighted sum program, it may be possible to increase the knowledge regarding the explored regions at each iteration. Larger regions would be eliminated by unfeasibility, while still maintaining the advantages of the proposed method.

Finally, the proposed DST has significant room for improvement, either by encompassing other variants of the LRP (or related problems), or by incorporating other functionality (e.g. integrating with geographic information systems). Also, the proposed interface solutions can be used to develop a browser-based application (or a Web service). Providing such an application may further improve the availability of the tool to DMs, while enabling to continuously update and maintain both the tool, as well as the used algorithms.

Nowadays, with the increasing interest and awareness of the importance of these problems, further development of DSTs may be a promising research subject, from where both practitioners and researchers can profit. Moreover, the need to divulge the works in the area to wider audiences (in order to reach DMs) as well as the recent appearance of new and extremely dynamic (and increasingly integrated) navigation tools further supports this interest.

As the best decision may not be the most cost efficient, this type of tool can help DMs make a more scientifically supported decision, by providing the total estimated costs of a set of different solutions (thus improving the DM’s insight and judgement). From this point on it is up to the manager to make the decision, taking into consideration other aspects such as the service level or even the organization strategy and motivation.

Overall, the work presented in this thesis may help to clarify some of the aspects related to the LRP, allowing to solve some of its variants, and presenting them in a way that researchers, DMs and the general public can use and contribute to.

References

Abran, A., Moore, J.W., Bourque, P., Dupuis, R. and Tripp, L.L. (ed.) (2004) *SWEBOK: Guide to the Software Engineering Body of Knowledge*, Los Alamitos: Institute of Electrical and Electronics Engineers.

Aksen, D. and Altinkemer, K. (2008) 'A location-routing problem for the conversion to the "click-and-mortar" retailing: the static case', *European Journal of Operational Research*, vol. 186, no. 2, pp. 554-575.

Albareda-Sambola, M. (2003) 'Models and Algorithms for Location-Routing and Related Problems', PhD thesis, Polytechnic University of Catalonia, Barcelona.

Albareda-Sambola, M., Díaz, J.A. and Fernández, E. (2005) 'A compact model and tight bounds for a combined location-routing problem', *Computers & Operations Research*, vol. 32, no. 3, pp. 407-428.

Albareda-Sambola, M., Fernández, E. and Laporte, G. (2007) 'Heuristic and lower bound for a stochastic location-routing problem', *European Journal of Operational Research*, vol. 179, no. 3, pp. 940-955.

Alumur, S. and Kara, B.Y. (2007) 'A new model for the hazardous waste location-routing problem', *Computers & Operations Research*, vol. 34, no. 5, pp. 1406-1423.

Alves, M.J. and Clímaco, J. (2000) 'An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound', *European Journal of Operational Research*, vol. 124, no. 3, pp. 478-494.

Alves, M.J. and Clímaco, J. (2007) 'A review of interactive methods for multiobjective integer and mixed-integer programming', *European Journal of Operational Research*, vol. 180, no. 1, pp. 99-115.

Alves, M.J. and Clímaco, J. (2009) 'Multi-objective mixed integer programming', in Floudas, C.A. and Pardalos, P.M. (ed.) *Encyclopedia of Optimization*, 2nd edition, New York: Springer.

Alves, M.J. and Costa, J.P. (2009) 'An exact method for computing the nadir values in multiple objective linear programming', *European Journal of Operational Research*, vol. 198, no. 2, pp. 637-646.

Amaya, A., Langevin, A. and Trépanier, M. (2007) 'The capacitated arc routing problem with refill points', *Operations Research Letters*, vol. 35, no. 1, pp. 45-53.

Ambler, S.W. (2002) *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*, New York: Wiley.

Ambrosino, D., Sciomachen, A. and Scutellà, M.G. (2009) 'A heuristic based on multi-exchange techniques for a regional fleet assignment location-routing problem', *Computers & Operations Research*, vol. 36, no. 2, pp. 442-460.

Ambrosino, D. and Scutellà, M.G. (2005) 'Distribution network design: new problems and related models', *European Journal of Operational Research*, vol. 165, no. 3, pp. 610-624.

Antunes, A.P. (1999) 'Location analysis helps manage solid waste in central Portugal', *Interfaces*, vol. 29, no. 4, pp. 32-43.

Antunes, A.P., Teixeira, J.C. and Coutinho, M.S. (2008) 'Managing solid waste through discrete location analysis: a case study in central Portugal', *Journal of the Operational Research Society*, vol. 59, no. 8, pp. 1038-1046.

Applegate, D.L., Bixby, R.E., Chvátal, V. and Cook, W.J. (2006) *The Traveling Salesman Problem: A Computational Study*, Princeton: Princeton University Press.

Averbakh, I. and Berman, O. (1994) 'Routing and location-routing p-delivery men problems on a path', *Transportation Science*, vol. 28, no. 2, pp. 162-166.

Averbakh, I. and Berman, O. (1995) 'Probabilistic sales-delivery man and sales-delivery facility location problems on a tree', *Transportation Science*, vol. 29, no. 2, pp. 184-197.

Averbakh, I. and Berman, O. (2002) 'Minmax p-traveling salesmen location problems on a tree', *Annals of Operations Research*, vol. 110, no. 1-4, pp. 55-68.

Averbakh, I., Berman, O. and Simchi-Levi, D. (1994) 'Probabilistic a priori routing-location problems', *Naval Research Logistics*, vol. 41, no. 7, pp. 973-989.

Aykin, T. (1995) 'The hub location and routing problem', *European Journal of Operational Research*, vol. 83, no. 1, pp. 200-219.

Balakrishnan, A., Ward, J.E. and Wong, R.T. (1987) 'Integrated facility location and vehicle routing models: recent work and future prospects', *American Journal of Mathematical and Management Sciences*, vol. 7, no. 1-2, pp. 35-61.

Baldacci, R. and Maniezzo, V. (2006) 'Exact methods based on node-routing formulations for undirected arc-routing problems', *Networks*, vol. 47, no. 1, pp. 52-60.

Balinski, M.L. (1965) 'Integer programming: methods, uses, computations', *Management Science*, vol. 12, no. 3, pp. 253-313.

Barreto, S., Ferreira, C., Paixão, J. and Santos, B.S. (2007) 'Using clustering analysis in a capacitated location-routing problem', *European Journal of Operational Research*, vol. 179, no. 3, pp. 968-977.

Beasley, J.E. and Nascimento, E.M. (1996) 'The vehicle routing-allocation problem: a unifying framework', *TOP*, vol. 4, no. 1, pp. 65-86.

Beck, K. (1999) 'Embracing change with extreme programming', *Computer*, vol. 32, no. 10, pp. 70-77.

Beck, K. and Andres, C. (2004) *Extreme Programming Explained: Embrace Change*, 2^{nd} edition, Reading: Addison-Wesley.

Belenguer, J.M., Benavent, E., Lacomme, P. and Prins, C. (2006) 'Lower and upper bounds for the mixed capacitated arc routing problem', *Computers & Operations Research*, vol. 33, no. 12, pp. 3363-3383.

Benavent, E., Campos, V., Corberán, Á. and Mota, E. (1992) 'The capacitated chinese postman problem: lower bounds', *Networks*, vol. 22, no. 7, pp. 669-690.

Berger, R.T., Coullard, C.R. and Daskin, M.S. (2007) 'Location-routing problems with distance constraints', *Transportation Science*, vol. 41, no. 1, pp. 29-43.

Berman, O. and Drezner, Z. (2000) 'A note on the location of an obnoxious facility on a network', *European Journal of Operational Research*, vol. 120, no. 1, pp. 215-217.

Berman, O., Drezner, Z. and Wesolowsky, G.O. (2003) 'The expropriation location problem', *Journal of the Operational Research Society*, vol. 54, no. 7, pp. 769-776.

Berman, O. and Krass, D. (2002) 'Facility location problems with stochastic demands and congestion', in Drezner, Z. and Hamacher, H.W. (ed.) *Facility Location: Applications and Theory*, Berlin: Springer-Verlag.

Berman, O. and Simchi-Levi, D. (1986) 'Minisum location of a traveling salesman', *Networks*, vol. 16, no. 3, pp. 239-254.

Berman, O. and Simchi-Levi, D. (1988a) 'Minisum location of a travelling salesman on simple networks', *European Journal of Operational Research*, vol. 36, no. 2, pp. 241-250.

Berman, O. and Simchi-Levi, D. (1988b) 'Finding the optimal a priori tour and location of a traveling salesman with nonhomogeneous customers', *Transportation Science*, vol. 22, no. 2, pp. 148-154.

Berman, O. and Simchi-Levi, D. (1989) 'The traveling salesman location problem on stochastic networks', *Transportation Science*, vol. 23, no. 1, pp. 54-57.

Berman, O., Simchi-Levi, D. and Tamir, A. (1988) 'The minimax multistop location problem on a tree', *Networks*, vol. 18, no. 1, pp. 39-49.

Berman, O. and Wang, Q. (2008) 'Locating a semi-obnoxious facility with expropriation', *Computers & Operations Research*, vol. 35, no. 2, pp. 392-403.

Bertsimas, D.J. (1989) 'Traveling salesman facility location problems', *Transportation Science*, vol. 23, no. 3, pp. 184-191.

Bertsimas, D.J., Jaillet, P. and Odoni, A.R. (1990) 'A priori optimization', *Operations Research*, vol. 38, no. 6, pp. 1019-1033.

Beullens, P., Muyldermans, L., Cattrysse, D. and Van Oudheusden, D. (2003) 'A guided local search heuristic for the capacitated arc routing problem', *European Journal of Operational Research*, vol. 147, no. 3, pp. 629-643.

Billionnet, A., Elloumi, S. and Grouz Djerbi, L. (2005) 'Designing radio-mobile access networks based on synchronous digital hierarchy rings', *Computers & Operations Research*, vol. 32, no. 2, pp. 379-394.

Blanchard, B.S. (2008) *System Engineering Management*, 4th edition, Hoboken: Wiley.

Blanquero, R. and Carrizosa, E. (2002) 'A d.c. biobjective location model', *Journal of Global Optimization*, vol. 23, no. 2, pp. 139-154.

Boffey, T.B. and Karkazis, J. (1993) 'Models and methods for location and routing decisions relating to hazardous materials', *Studies in Locational Analysis*, vol. 5, pp. 149-166.

Boffey, T.B. and Karkazis, J. (1995) 'Location, routing and the environment', in Drezner, Z. (ed.) *Facility Location: A Survey of Applications and Methods*, New York: Springer-Verlag.

Boffey, T.B., Mesa, J.A., Ortega, F.A. and Rodrigues, J.I. (2008) 'Locating a low-level waste disposal site', *Computers & Operations Research*, vol. 35, no. 3, pp. 701-716.

Booch, G., Rumbaugh, J. and Jacobson, I. (2005) *The Unified Modeling Language User Guide*, 2nd edition, Reading: Addison-Wesley.

Bookbinder, J.H. and Reece, K.E. (1988) 'Vehicle routing considerations in distribution system design', *European Journal of Operational Research*, vol. 37, no. 2, pp. 204-213.

Box, G.E.P., Hunter, J.S. and Hunter, W.G. (2005) *Statistics for Experimenters: Design, Innovation, and Discovery*, 2nd edition, Hoboken: Wiley.

Bramel, J. and Simchi-Levi, D. (1997) *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*, New York: Springer-Verlag.

Branco, I.M. and Coelho, J.D. (1990) 'The Hamiltonian p-median problem', *European Journal of Operational Research*, vol. 47, no. 1, pp. 86-95.

Brimberg, J. and Juel, H. (1998) 'A bicriteria model for locating a semi-desirable facility in the plane', *European Journal of Operational Research*, vol. 106, no. 1, pp. 144-151.

Brimberg, J. and Mehrez, A. (1994) 'Multi-facility location using a maximin criterion and rectangular distances', *Location Science*, vol. 2, no. 1, pp. 11-19.

Brimberg, J. and Wesolowsky, G.O. (1995) 'The rectilinear distance minisum problem with minimum distance constraints', *Location Science*, vol. 3, no. 3, pp. 203-215.

Bruns, A., Klose, A. and Stähly, P. (2000) 'Restructuring of Swiss parcel delivery services', *OR Spektrum*, vol. 22, no. 2, pp. 285-302.

Burkard, R.E., Fathali, J. and Kakhki, H.T. (2007) 'The p-maxian problem on a tree', *Operations Research Letters*, vol. 35, no. 3, pp. 331-335.

Burness, R.C. and White, J.A. (1976) 'The traveling salesman location problem', *Transportation Science*, vol. 10, no. 4, pp. 348-360.

Burstein, F. and Holsapple, C.W. (ed.) (2008) *Handbook on Decision Support Systems 1: Basic Themes*, Berlin: Springer-Verlag.

Caballero, R., González, M., Guerrero, F.M., Molina, J. and Paralera, C. (2007) 'Solving a multiobjective location routing problem with a metaheuristic based on tabu search. Application to a real case in Andalusia', *European Journal of Operational Research*, vol. 177, no. 3, pp. 1751-1763.

Cappanera, P. (1999) 'Discrete Facility Location and Routing of Obnoxious Activities', PhD thesis, University of Milan, Milan.

Cappanera, P., Gallo, G. and Maffioli, F. (2004) 'Discrete facility location and routing of obnoxious activities', *Discrete Applied Mathematics*, vol. 133, no. 1-3, pp. 3-28.

Carrizosa, E. and Conde, E. (2002) 'A fractional model for locating semi-desirable facilities on networks', *European Journal of Operational Research*, vol. 136, no. 1, pp. 67-80.

Carrizosa, E. and Plastria, F. (1999) 'Location of semi-obnoxious facilities', *Studies in Locational Analysis*, vol. 12, pp. 1-27.

Chan, Y. and Baker, S.F. (2005) 'The multiple depot, multiple traveling salesmen facility-location problem: vehicle range, service frequency, and heuristic implementations', *Mathematical and Computer Modelling*, vol. 41, no. 8-9, pp. 1035-1053.

Chan, Y., Carter, W.B. and Burnes, M.D. (2001) 'A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands', *Computers & Operations Research*, vol. 28, no. 8, pp. 803-826.

Chandrasekaran, R. and Daughety, A. (1981) 'Location on tree networks: p-centre and n-dispersion problems', *Mathematics of Operations Research*, vol. 6, no. 1, pp. 50-57.

Chan, A.W. and Francis, R.L. (1976) 'A round-trip location problem on a tree graph', *Transportation Science*, vol. 10, no. 1, pp. 35-51.

Chan, A.W. and Hearn, D.W. (1977) 'A rectilinear distance round-trip location problem', *Transportation Science*, vol. 11, no. 2, pp. 107-123.

Chao, I.-M. (2002) 'A tabu search method for the truck and trailer routing problem', *Computers & Operations Research*, vol. 29, no. 1, pp. 33-51.

Chaudhry, S.S. (2006) 'A genetic algorithm approach to solving the anti-covering location problem', *Expert Systems*, vol. 23, no. 5, pp. 251-257.

Chaudhry, S.S., McCormick, S.T. and Moon, I.D. (1986) 'Locating independent facilities with maximum weight: greedy heuristics', *Omega*, vol. 14, no. 5, pp. 383-389.

Chien, T.W. (1993) 'Heuristic procedures for practical-sized uncapacitated location-capacitated routing problems', *Decision Sciences*, vol. 24, no. 5, pp. 995-1021.

Church, R.L. and Garfinkel, R.S. (1978) 'Locating an obnoxious facility on a network', *Transportation Science*, vol. 12, no. 2, pp. 107-118.

Church, R.L. and ReVelle, C.S. (1974) 'The maximal covering location problem', *Papers of the Regional Science Association*, vol. 32, no. 1, pp. 101-118.

Clarke, G. and Wright, J. (1964) 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research*, vol. 12, no. 4, pp. 568-581.

Clímaco, J., Ferreira, C. and Captivo, M.E. (1997) 'Multicriteria integer programming: an overview of the different algorithmic approaches', in Clímaco, J. (ed.) *Multicriteria Analysis*, Berlin: Springer-Verlag.

Coello Coello, C.A., Lamont, G.B. and Van Veldhuizen, D.A. (2007) *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edition, New York: Springer.

Cohn, M. (2004) *User Stories Applied: For Agile Software Development*, Boston: Addison-Wesley.

Colebrook, M., Gutierrez, J., Alonso, S. and Sicilia, J. (2002) 'A new algorithm for the undesirable 1-center problem on networks', *Journal of the Operational Research Society*, vol. 53, no. 12, pp. 1357-1366.

Colebrook, M. and Sicilia, J. (2007) 'Undesirable facility location problems on multicriteria networks', *Computers & Operations Research*, vol. 34, no. 5, pp. 1491-1514.

Cooper, L. (1972) 'The transportation-location problem', *Operations Research*, vol. 20, no. 1, pp. 94-108.

Cooper, L. (1976) 'An efficient heuristic algorithm for the transportation-location problem', *Journal of Regional Science*, vol. 16, no. 3, pp. 309-315.

Cooper, L. (1978) 'The stochastic transportation-location problem', *Computers & Mathematics with Applications*, vol. 4, no. 3, pp. 265-275.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M.M. and Soumis, F. (2002) 'VRP with time windows', in Toth, P. and Vigo, D. (ed.) *The Vehicle Routing Problem*, Philadelphia: Society for Industrial and Applied Mathematics.

Coutinho-Rodrigues, J., Current, J., Clímaco, J. and Ratick, S. (1997) 'Interactive spatial decision-support system for multiobjective hazardous materials location-routing problems', *Transportation Research Record*, no. 1602, pp. 101-109.

Crainic, T.G. and Laporte, G. (ed.) (1998) *Fleet Management and Logistics*, Norwell: Kluwer Academic Publishers.

Current, J., Daskin, M. and Schilling, D. (2002) 'Discrete network location models', in Drezner, Z. and Hamacher, H.W. (ed.) *Facility Location: Applications and Theory*, Berlin: Springer-Verlag.

Current, J. and Ratick, S. (1995) 'A model to assess risk, equity and efficiency in facility location and transportation of hazardous materials', *Location Science*, vol. 3, no. 3, pp. 187-201.

Daganzo, C.F. (2005) *Logistics Systems Analysis*, 4th edition, Berlin: Springer-Verlag.

Dantzig, G.B. and Ramser, J.H. (1959) 'The truck dispatching problem', *Management Science*, vol. 6, no. 1, pp. 80-91.

Dasarathy, B. and White, L.J. (1980) 'A maximin location problem', *Operations Research*, vol. 28, no. 6, pp. 1385-1401.

Daskin, M.S. (1995) *Network and Discrete Location: Models, Algorithms, and Applications*, New York: Wiley-Interscience.

Deb, K. (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*, Chichester: Wiley.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002) 'A fast elitist multi-objective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197.

Demis (2010) *Demis products: Web Map Server* [Online], Available: http://www.demis.nl/home/pages/wms/demiswms.htm [25 Nov 2010].

Dijkstra, E.W. (1959) 'A note on two problems in connexion with graphs', *Numerische Mathematik*, vol. 1, no. 1, pp. 269-271.

Dix, A., Finlay, J.E., Abowd, G.D. and Beale, R. (2004) *Human-Computer Interaction*, 3rd edition, Harlow: Pearson-Prentice Hall.

Drezner, Z. (1982) 'Fast algorithms for the round trip location problem', *IEE Transactions*, vol. 14, no. 4, pp. 243-248.

Drezner, Z. (1985) 'O(N log N) algorithm for the rectilinear round-trip location problem', *Transportation Science*, vol. 19, no. 1, pp. 91-100.

Drezner, Z. and Erkut, E. (1995) 'Solving the continuous p-dispersion problem using non-linear programming', *Journal of the Operational Research Society*, vol. 46, no. 4, pp. 516-520.

Drezner, Z. and Hamacher, H.W. (ed.) (2002) *Facility Location: Applications and Theory*, Berlin: Springer-Verlag.

Drezner, Z., Steiner, G. and Wesolowsky, G.O. (1985) 'One-facility location with rectilinear tour distances', *Naval Research Logistics Quarterly*, vol. 32, no. 3, pp. 391-405.

Drezner, Z. and Wesolowsky, G.O. (1980) 'A maximin location problem with maximum distance constraints', *IIE Transactions*, vol. 12, no. 3, pp. 249-252.

Drezner, Z. and Wesolowsky, G.O. (1982) 'A trajectory approach for the round-trip location problem', *Transportation Science*, vol. 16, no. 1, pp. 56-66.

Drezner, Z. and Wesolowsky, G.O. (1985) 'Location of multiple obnoxious facilities', *Transportation Science*, vol. 19, no. 3, pp. 193-202.

Dror, M. (ed.) (2000) *Arc Routing: Theory, Solutions and Applications*, Norwell: Kluwer Academic Publishers.

Dugardin, F., Yalaoui, F. and Amodeo, L. (2010) 'New multi-objective method to solve reentrant hybrid flow shop scheduling problem', *European Journal of Operational Research*, vol. 203, no. 1, pp. 22-31.

Duhamel, C., Lacomme, P., Prins, C. and Prodhon, C. (2010) 'A GRASPxELS approach for the capacitated location-routing problem', *Computers & Operations Research*, vol. 37, no. 11, pp. 1912-1923.

Durso, A. (1992) 'An interactive combined branch-and-bound/Tchebycheff algorithm for multiple criteria optimization', in Goicoechea, A., Duckstein, L. and Zionts, S. (ed.) *Multiple*

Criteria Decision Making: Theory and Applications in Business, Industry and Government, New York: Springer-Verlag.

Eglese, R.W. (1994) 'Routeing winter gritting vehicles', *Discrete Applied Mathematics*, vol. 48, no. 3, pp. 231-244.

Ehrgott, M. and Gandibleux, X. (2002) 'Multiobjective combinatorial optimization — theory, methodology, and applications', in Ehrgott, M. and Gandibleux, X. (ed.) *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, Norwell: Kluwer Academic Publishers.

Ehrgott, M. and Verma, R. (2001) 'A note on solving multicriteria transportation-location problems by fuzzy programming', *Asia-Pacific Journal of Operational Research*, vol. 18, no. 2, pp. 149-164.

Eiselt, H.A. and Laporte, G. (1995) 'Objectives in location problems', in Drezner, Z. (ed.) *Facility Location: A Survey of Applications and Methods*, New York: Springer-Verlag.

Eom, S.B. and Kim, E.B. (2006) 'A survey of decision support system applications (1995-2001)', *Journal of the Operational Research Society*, vol. 57, no. 11, pp. 1264-1278.

Eom, H.B. and Lee, S.M. (1990) 'A survey of decision support system applications (1971-April 1988)', *Interfaces*, vol. 20, no. 3, pp. 65-79.

Eom, S.B., Lee, S.M., Kim, E.B. and Somarajan, C. (1998) 'A survey of decision support system applications (1988-1994)', *Journal of the Operational Research Society*, vol. 49, no. 2, pp. 109-120.

Erkut, E. (1990) 'The discrete p-dispersion problem', *European Journal of Operational Research*, vol. 46, no. 1, pp. 48-60.

Erkut, E., Baptie, T. and Von Hohenbalken, B. (1990) 'The discrete p-Maxian location problem', *Computers & Operations Research*, vol. 17, no. 1, pp. 51-61.

Erkut, E. and Neuman, S. (1989) 'Analytical models for locating undesirable facilities', *European Journal of Operational Research*, vol. 40, no. 3, pp. 275-291.

Erkut, E. and Neuman, S. (1992) 'A multiobjective model for locating undesirable facilities', *Annals of Operations Research*, vol. 40, no. 1, pp. 209-227.

Erkut, E. and Öncü, T.S. (1991) 'A parametric 1-maximin location problem', *Journal of the Operational Research Society*, vol. 42, no. 1, pp. 49-55.

Erkut, E., ReVelle, C.S. and Ülküsal, Y. (1996) 'Integer-friendly formulations for the r-separation problem', *European Journal of Operational Research*, vol. 92, no. 2, pp. 342-351.

Erkut, E. and Verter, V. (1995) 'Hazardous materials logistics', in Drezner, Z. (ed.) *Facility Location: A Survey of Applications and Methods*, New York: Springer-Verlag.

Evans, G.W. (1984) 'An overview of techniques for solving multiobjective mathematical programs', *Management Science*, vol. 30, no. 11, pp. 1268-1282.

Ferreira, C. (1997) 'Problemas de localização e distribuição multicritério — aproximações e estudo de alguns casos com implicações ambientais', PhD thesis, University of Aveiro, Aveiro [in Portuguese].

Ferreira, C., Lopes, R.B. and Santos, B.S. (2010) 'An interactive method for multi-objective integer and mixed-integer programming applied to a facility location problem', 24th European Conference on Operational Research, Lisbon, 160.

Ferreira, C., Santos, B.S., Captivo, M.E., Clímaco, J. and Silva, C.C. (1996) 'Multiobjective location of unwelcome or central facilities involving environmental aspects – a prototype of a decision support system', *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 36, no. 2-3, pp. 159-172.

Filho, V.J.M.F. and Galvão, R.D. (1998) 'A tabu search heuristic for the concentrator location problem', *Location Science*, vol. 6, no. 1, pp. 189-209.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W. H. Freeman.

Gerdessen, J.C. (1996) 'Vehicle routing problem with trailers', *European Journal of Operational Research*, vol. 93, no. 1, pp. 135-147.

Ghiani, G. and Improta, G. (2000) 'An efficient transformation of the generalized vehicle routing problem', *European Journal of Operational Research*, vol. 122, no. 1, pp. 11-17.

Ghiani, G. and Laporte, G. (1999) 'Eulerian location problems', *Networks*, vol. 34, no. 4, pp. 291-302.

Ghiani, G. and Laporte, G. (2001) 'Location-arc routing problems', *Opsearch*, vol. 38, no. 2, pp. 151-159.

Ghiani, G., Laporte, G. and Musmanno, R. (2004) *Introduction to Logistics Systems Planning and Control*, Chichester: Wiley.

Ghosh, J.B. (1996) 'Computational aspects of the maximum diversity problem', *Operations Research Letters*, vol. 19, no. 4, pp. 175-181.

Giannikos, I. (1993) 'Locating multiple obnoxious facilities in the plane under the rectilinear metric', *Studies in Locational Analysis*, vol. 4, pp. 71-73.

Giannikos, I. (1998) 'A multiobjective programming model for locating treatment sites and routing hazardous wastes', *European Journal of Operational Research*, vol. 104, no. 2, pp. 333-342.

Glover, F. (1986) 'Future paths for integer programming and links to artificial intelligence', *Computers & Operations Research*, vol. 13, no. 5, pp. 533-549.

Glover, F.W. and Kochenberger, G.A. (ed.) (2003) *Handbook of Metaheuristics*, Norwell: Kluwer Academic Publishers.

Goetschalckx, M., Vidal, C.J. and Dogan, K. (2002) 'Modeling and design of global logistics systems: a review of integrated strategic and tactical models and design algorithms', *European Journal of Operational Research*, vol. 143, no. 1, pp. 1-18.

Golden, B.L., DeArmon, J.S. and Baker, E.K. (1983) 'Computational experiments with algorithms for a class of routing problems', *Computers & Operations Research*, vol. 10, no. 1, pp. 47-59.

Golden, B.L. and Wong, R.T. (1981) 'Capacitated arc routing problems', *Networks*, vol. 11, no. 3, pp. 305-315.

Gordillo, J., Plastria, F. and Carrizosa, E. (2006) 'Locating a semi-obnoxious facility with repelling polygonal regions', MOSI Department, Vrije Universiteit Brussel, Brussel.

Gorr, W.L., Johnson, M.P. and Roehrig, S.F. (2001) 'Spatial decision support system for home-delivered services', *Journal of Geographical Systems*, vol. 3, no. 2, pp. 181-197.

Gorry, G.A. and Scott-Morton, M.S. (1971) 'A framework for management information systems', *Sloan Management Review*, vol. 13, no. 1, pp. 55-70.

Guerra, L., Murino, T. and Romano, E. (2007) 'A heuristic algorithm for the constrained location - routing problem', *International Journal of Systems Applications, Engineering & Development*, vol. 1, no. 4, pp. 146-154.

Gunnarsson, H., Rönnqvist, M. and Carlsson, D. (2006) 'A combined terminal location and ship routing problem', *Journal of the Operational Research Society*, vol. 57, no. 8, pp. 928-938.

Hair, J.F., Black, W.C., Babin, B.J. and Anderson, R.E. (2009) *Multivariate Data Analysis*, 7th edition, Upper Saddle River: Prentice Hall.

Hakimi, S.L. (1964) 'Optimum locations of switching centers and the absolute centers and medians of a Graph', *Operations Research*, vol. 12, no. 3, pp. 450-459.

Halpern, J. (1978) 'Finding minimal center-median convex combination (cent-dian) of a graph', *Management Science*, vol. 24, no. 5, pp. 535-544.

Hamacher, H.W., Labbé, M., Nickel, S. and Skriver, A.J.V. (2002) 'Multicriteria semi-obnoxious network location problems (MSNLP) with sum and center objectives', *Annals of Operations Research*, vol. 110, no. 1-4, pp. 33-53.

Hansen, P.H., Hegedahl, B., Hjortkjær, S. and Obel, B. (1994) 'A heuristic solution to the warehouse location-routing problem', *European Journal of Operational Research*, vol. 76, no. 1, pp. 111-127.

Hansen, P. and Mladenović, N. (2001) 'Variable neighborhood search: principles and applications', *European Journal of Operational Research*, vol. 130, no. 3, pp. 449-467.

Hansen, P. and Mladenović, N. (2006) 'First vs. best improvement: an empirical study', *Discrete Applied Mathematics*, vol. 154, no. 5, pp. 802-817.

Hansen, P., Peeters, D., Richard, D. and Thisse, J.-F. (1985) 'The minisum and minimax location problems revisited', *Operations Research*, vol. 33, no. 6, pp. 1251-1265.

Hansen, P., Peeters, D. and Thisse, J.-F. (1981) 'On the location of an obnoxious facility', *Sistemi Urbani*, vol. 3, pp. 299-317.

Hertz, A. and Widmer, M. (2003) 'Guidelines for the use of meta-heuristics in combinatorial optimization', *European Journal of Operational Research*, vol. 151, no. 2, pp. 247-252.

Highsmith, J. and Cockburn, A. (2001) 'Agile software development: the business of innovation', *Computer*, vol. 34, no. 9, pp. 120-122.

Hindi, K.S. and Basta, T. (1994) 'Computationally efficient solution of a multiproduct, two-stage distribution-location problem', *Journal of the Operational Research Society*, vol. 45, no. 11, pp. 1316-1323.

Hoaglin, D.C., Mosteller, F. and Tukey, J.W. (ed.) (2000) *Understanding Robust and Exploratory Data Analysis*, New York: Wiley.

Holsapple, C.W. (2008) 'DSS architecture and types', in Burstein, F. and Holsapple, C.W. (ed.) *Handbook on Decision Support Systems 1: Basic Themes*, Berlin: Springer-Verlag.

House, R.G. and Karrenbauer, J.J. (1982) 'Logistics system modelling', *International Journal of Physical Distribution & Logistics Management*, vol. 12, no. 3, pp. 119-129.

Isermann, H. and Steuer, R.E. (1988) 'Computational experience concerning payoff tables and minimum criterion values over the efficient set', *European Journal of Operational Research*, vol. 33, no. 1, pp. 91-97.

ISO (1998) *ISO 9241-11 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability*, Geneva: International Organization for Standardization.

Ivory, M.Y. and Hearst, M.A. (2001) 'The state of the art in automating usability evaluation of user interfaces', *ACM Computing Surveys*, vol. 33, no. 4, pp. 470-516.

Jacobsen, S.K. and Madsen, O.B.G. (1980) 'A comparative study of heuristics for a two-level routing-location problem', *European Journal of Operational Research*, vol. 5, no. 6, pp. 378-387.

Jacobson, I., Booch, G. and Rumbaugh, J. (1999) *The Unified Software Development Process*, Reading: Addison-Wesley.

Jacobs, T.L. and Warmerdam, J.M. (1994) 'Simultaneous routing and siting for hazardous-waste operations', *Journal of Urban Planning and Development*, vol. 120, no. 3, pp. 115-131.

Jaszkiewicz, A. (2004) 'Evaluation of multiple objective metaheuristics', in Gandibleux, X., Sevaux, M., Sörensen, K. and T'kindt, V. (ed.) *Metaheuristics for Multiobjective Optimisation*, Berlin: Springer-Verlag.

Johnson, M.P., Gorr, W.L. and Roehrig, S.F. (2002) 'Location/allocation/routing for home-delivered meals provision', *International Journal of Industrial Engineering*, vol. 9, no. 1, pp. 45-56.

Karaivanova, J., Korhonen, P., Narula, S.C., Wallenius, J. and Vassilev, V. (1995) 'A reference direction approach to multiple objective integer linear programming', *European Journal of Operational Research*, vol. 81, no. 1, pp. 176-187.

Karasakal, E. and Nadirler, D. (2008) 'An interactive solution approach for a bi-objective semi-desirable location problem', *Journal of Global Optimization*, vol. 42, no. 2, pp. 177-199.

Karkazis, J. and Karagiorgis, P. (1986) 'A method to locate the maximum circle(s) inscribed in a polygon', *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 26, no. 3, pp. 3-36.

Karkazis, J. and Karagiorgis, P. (1988) 'An algorithm solving the general problem of locating obnoxious facilities on the plane', *Operational Research '87*, pp. 702-717.

Keen, P.G.W. and Scott-Morton, M.S. (1978) *Decision Support Systems: An Organizational Perspective*, Reading: Addison-Wesley.

Kincaid, R.K. (1992) 'Good solutions to discrete noxious location problems via metaheuristics', *Annals of Operations Research*, vol. 40, no. 1, pp. 265-281.

Kincaid, R.K. and Yellin, L.G. (1993) 'The discrete p-dispersion-sum problem: results on trees and graphs', *Location Science*, vol. 1, no. 2, pp. 171-186.

Klee, M. (2000) *Five paper prototyping tips* [Online], Available: http://www.uie.com/articles/prototyping_tips/ [25 Nov 2010].

Klein, C.M. and Kincaid, R.K. (1994) 'The discrete anti-p-center problem', *Transportation Science*, vol. 28, no. 1, pp. 77-79.

Kolen, A. (1985) 'The round-trip p-center and covering problem on a tree', *Transportation Science*, vol. 19, no. 3, pp. 222-234.

Korhonen, P. and Wallenius, J. (1988) 'A pareto race', *Naval Research Logistics*, vol. 35, no. 6, pp. 615-623.

Krarup, J., Pisinger, D. and Plastria, F. (2002) 'Discrete location problems with push–pull objectives', *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 363-378.

Kulcar, T. (1996) 'Optimizing solid waste collection in Brussels', *European Journal of Operational Research*, vol. 90, no. 1, pp. 71-77.

Kvam, P.H. and Vidakovic, B. (2007) *Nonparametric Statistics with Applications to Science and Engineering*, Hoboken: Wiley.

Labbé, M. and Laporte, G. (1986) 'Maximizing user convenience and postal service efficiency in post box location', *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 26, no. 2, pp. 21-35.

Labbé, M., Laporte, G., Rodríguez Martín, I. and Salazar González, J.J. (2005) 'Locating median cycles in networks', *European Journal of Operational Research*, vol. 160, no. 2, pp. 457-470.

Labbé, M., Rodríguez Martín, I. and Salazar González, J.J. (2004) 'A branch-and-cut algorithm for the plant-cycle location problem', *Journal of the Operational Research Society*, vol. 55, no. 5, pp. 513-520.

Lacomme, P., Prins, C. and Sevaux, M. (2006) 'A genetic algorithm for a bi-objective capacitated arc routing problem', *Computers & Operations Research*, vol. 33, no. 12, pp. 3473-3493.

Laporte, G. (1988) 'Location-routing problems', in Golden, B.L. and Assad, A.A. (ed.) *Vehicle Routing: Methods and Studies*, Amsterdam: North-Holland.

Laporte, G. (1989) 'A survey of algorithms for location-routing problems', *Investigación Operativa*, vol. 1, no. 2, pp. 93-123.

Laporte, G. and Dejax, P.J. (1989) 'Dynamic location-routeing problems', *Journal of the Operational Research Society*, vol. 40, no. 5, pp. 471-482.

Laporte, G., Louveaux, F. and Mercure, H. (1989) 'Models and exact solutions for a class of stochastic location-routing problems', *European Journal of Operational Research*, vol. 39, no. 1, pp. 71-78.

Laporte, G. and Nobert, Y. (1981) 'An exact algorithm for minimizing routing and operating costs in depot location', *European Journal of Operational Research*, vol. 6, no. 2, pp. 224-226.

Laporte, G. and Nobert, Y. (1983) 'Generalized travelling salesman problem through n sets of nodes: an integer programming approach', *INFOR*, vol. 21, no. 1, pp. 61-75.

Laporte, G., Nobert, Y. and Arpin, D. (1986) 'An exact algorithm for solving a capacitated location-routing problem', *Annals of Operations Research*, vol. 6, no. 9, pp. 293-310.

Laporte, G., Nobert, Y. and Pelletier, P. (1983) 'Hamiltonian location problems', *European Journal of Operational Research*, vol. 12, no. 1, pp. 82-89.

Laporte, G., Nobert, Y. and Taillefer, S. (1988) 'Solving a family of multi-depot vehicle routing and location-routing problems', *Transportation Science*, vol. 22, no. 3, pp. 161-172.

Lashine, S.H., Fattouh, M. and Issa, A. (2006) 'Location/allocation and routing decisions in supply chain network design', *Journal of Modelling in Management*, vol. 1, no. 2, pp. 173-183.

Lawrence, R.M. and Pengilly, P.J. (1969) 'The number and location of depots required for handling products for distribution to retail stores in south-east England', *Operational Research Quarterly*, vol. 20, no. 1, pp. 23-32.

LeBlanc, L.J. (1977) 'A heuristic approach for large scale discrete stochastic transportation-location problems', *Computers & Mathematics with Applications*, vol. 3, no. 2, pp. 87-94.

Lee, Y., Kim, S.-I., Lee, S. and Kang, K. (2003) 'A location-routing problem in designing optical internet access with WDM systems', *Photonic Network Communications*, vol. 6, no. 2, pp. 151-160.

Levy, L. and Bodin, L. (1989) 'The arc oriented location routing problem', *INFOR*, vol. 27, no. 1, pp. 74-94.

L'Hoir, H. and Teghem, J. (1995) 'Portfolio selection by MOLP using an interactive branch and bound', *Foundations of Computing and Decision Sciences*, vol. 20, no. 3, pp. 175-185.

Lin, C.K.Y., Chow, C.K. and Chen, A. (2002) 'A location-routing-loading problem for bill delivery services', *Computers & Industrial Engineering*, vol. 43, no. 1-2, pp. 5-25.

Lin, S. and Kernighan, B.W. (1973) 'An effective heuristic algorithm for the traveling salesman problem', *Operations Research*, vol. 21, no. 2, pp. 498-516.

Lin, C.K.Y. and Kwok, R.C.W. (2006) 'Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data', *European Journal of Operational Research*, vol. 175, no. 3, pp. 1833-1849.

Lin, J.-R. and Lei, H.-C. (2009) 'Distribution systems design with two-level routing considerations', *Annals of Operations Research*, vol. 172, no. 1, pp. 329-347.

List, G.F. and Mirchandani, P.B. (1991) 'An integrated network/planar multiobjective model for routing and siting for hazardous materials and wastes', *Transportation Science*, vol. 25, no. 2, pp. 146-156.

Liu, S.C. and Lee, S.B. (2003) 'A two-phase heuristic method for the multi-depot location routing problem taking inventory control decisions into considerations', *International Journal of Advanced Manufacturing Technology*, vol. 22, no. 11, pp. 941-950.

Liu, S.C. and Lin, C.C. (2005) 'A heuristic method for the combined location routing and inventory problem', *International Journal of Advanced Manufacturing Technology*, vol. 26, no. 4, pp. 372-381.

Longo, H., Aragão, M.P.d. and Uchoa, E. (2006) 'Solving capacitated arc routing problems using a transformation to the CVRP', *Computers & Operations Research*, vol. 33, no. 6, pp. 1823-1837.

Lopes, R.B., Barreto, S., Ferreira, C. and Santos, B.S. (2008a) 'A decision-support tool for a capacitated location-routing problem', *Decision Support Systems*, vol. 46, no. 1, pp. 366-375.

Lopes, R.B., Ferreira, C. and Santos, B.S. (2009) 'Solving the capacitated location-routing problem by a guided local search metaheuristic', VIII Metaheuristic International Conference, Hamburg, 7 pages (in CD-ROM).

Lopes, R.B., Ferreira, C. and Santos, B.S. (2010a) 'A multi-objective evolutionary algorithm for the capacitated location-routing problem', 24th European Conference on Operational Research, Lisbon, 160.

Lopes, R.B., Ferreira, C., Santos, B.S. and Barreto, S. (2008b) 'A taxonomical analysis, current methods and objectives on location-routing problems', XVII International Meeting on Locational Analysis, Elche, 36.

Lopes, R.B., Plastria, F., Ferreira, C. and Santos, B.S. (2010b) 'Location-arc routing problems: heuristic approaches and test instances', XVIII International Meeting on Locational Analysis, Naples, 70.

Lozano, A.J. and Mesa, J.A. (2000) 'Location of facilities with undesirable effects and inverse location problems: a classification', *Studies in Locational Analysis*, vol. 14, pp. 253-291.

Madsen, O.B.G. (1981) 'A survey of methods for solving combined location-routing problems', in Jaiswal, N.K. (ed.) *Scientific Management of Transport Systems*, Amsterdam: North-Holland.

Madsen, O.B.G. (1983) 'Methods for solving combined two level location-routing problems of realistic dimensions', *European Journal of Operational Research*, vol. 12, no. 3, pp. 295-301.

Marakas, G.M. (1998) *Decision Support Systems in the 21st Century*, Upper Saddle River: Prentice Hall.

Maranzana, F.E. (1964) 'On the location of supply points to minimize transport costs', *Operational Research Quarterly*, vol. 15, no. 3, pp. 261-270.

Marcotte, O. and Soland, R.M. (1986) 'An interactive branch-and-bound algorithm for multiple criteria optimization', *Management Science*, vol. 32, no. 1, pp. 61-75.

Marinakis, Y. and Marinaki, M. (2008a) 'A bilevel genetic algorithm for a real location routing problem', *International Journal of Logistics Research and Applications*, vol. 11, no. 1, pp. 49-65.

Marinakis, Y. and Marinaki, M. (2008b) 'A particle swarm optimization algorithm with path relinking for the location routing problem', *Journal of Mathematical Modelling and Algorithms*, vol. 7, no. 1, pp. 59-78.

Mayhew, D.J. (1992) *Principles and Guidelines in Software User Interface Design*, Upper Saddle River: Prentice Hall.

McConnell, S. (1996) *Rapid Development: Taming Wild Software Schedules*, Redmond: Microsoft Press.

McDiarmid, C. (1992) 'Probability modelling and optimal location of a travelling salesman', *Journal of the Operational Research Society*, vol. 43, no. 5, pp. 533-538.

Melachrinoudis, E. (1999) 'Bicriteria location of a semi-obnoxious facility', *Computers & Industrial Engineering*, vol. 37, no. 3, pp. 581-593.

Melachrinoudis, E. and Cullinane, T.P. (1985) 'Locating an undesirable facility within a geographical region using the maximin criterion', *Journal of Regional Science*, vol. 25, no. 1, pp. 115-127.

Melachrinoudis, E. and Cullinane, T.P. (1986) 'Locating an undesirable facility with a minimax criterion', *European Journal of Operational Research*, vol. 24, no. 2, pp. 239-246.

Melachrinoudis, E., Min, H. and Wu, X. (1995) 'A multiobjective model for the dynamic location of landfills', *Location Science*, vol. 3, no. 3, pp. 143-166.

Melachrinoudis, E. and Smith, J.M. (1995) 'An $O(mn^2)$ algorithm for the maximin problem in E^2', *Operations Research Letters*, vol. 18, no. 1, pp. 25-30.

Melachrinoudis, E. and Xanthopulos, Z. (2003) 'Semi-obnoxious single facility location in Euclidean space', *Computers & Operations Research*, vol. 30, no. 14, pp. 2191-2209.

Melachrinoudis, E. and Zhang, F.G. (1999) 'An O(mn) algorithm for the 1-maximin problem on a network', *Computers & Operations Research*, vol. 26, no. 9, pp. 849-869.

Melechovský, J., Prins, C. and Wolfler Calvo, R. (2005) 'A metaheuristic to solve a location-routing problem with non-linear costs', *Journal of Heuristics*, vol. 11, no. 5, pp. 375-391.

Melo, M.T., Nickel, S. and Saldanha-da-Gama, F. (2009) 'Facility location and supply chain management – a review', *European Journal of Operational Research*, vol. 196, no. 2, pp. 401-412.

Mester, D. and Bräysy, O. (2005) 'Active guided evolution strategies for large-scale vehicle routing problems with time windows', *Computers & Operations Research*, vol. 32, no. 6, pp. 1593-1614.

Microsoft Corporation (2010) *Windows user experience interaction guidelines* [Online], Available: http://msdn.microsoft.com/en-us/library/aa511258(v=MSDN.10).aspx [25 Nov 2010].

Miettinen, K., Ruiz, F. and Wierzbicki, A.P. (2008) 'Introduction to multiobjective optimization: interactive approaches', in Branke, J., Deb, K., Miettinen, K. and Słowiński, R. (ed.) *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Berlin: Springer-Verlag.

Miller, G.A. (1956) 'The magical number seven, plus or minus two: some limits on our capacity for processing information', *Psychological Review*, vol. 63, no. 2, pp. 81-97.

Mills, P., Tsang, E. and Ford, J. (2003) 'Applying an extended guided local search to the quadratic assignment problem', *Annals of Operations Research*, vol. 118, no. 1-4, pp. 121-135.

Min, H. (1996) 'Consolidation terminal location-allocation and consolidated routing problems', *Journal of Business Logistics*, vol. 17, no. 2, pp. 235-263.

Min, H. and Eom, S.B. (1994) 'An integrated decision support system for global logistics', *International Journal of Physical Distribution & Logistics Management*, vol. 24, no. 1, pp. 29-39.

Minieka, E. (1983) 'Anticenters and antimedians of a network', *Networks*, vol. 13, no. 3, pp. 359-364.

Min, H., Jayaraman, V. and Srivastava, R. (1998) 'Combined location-routing problems: a synthesis and future research directions', *European Journal of Operational Research*, vol. 108, no. 1, pp. 1-15.

Mitchell, P.P. (2007) *A Step-by-Step Guide to Usability Testing*, Lincoln: iUniverse.

Mladenović, N. and Hansen, P. (1997) 'Variable neighborhood search', *Computers & Operations Research*, vol. 24, no. 11, pp. 1097-1100.

Moon, I.D. and Chaudhry, S.S. (1984) 'An analysis of network location problems with distance constraints', *Management Science*, vol. 30, no. 3, pp. 290-307.

Mosheiov, G. (1995) 'The pickup delivery location problem on networks', *Networks*, vol. 26, no. 4, pp. 243-251.

Muñoz-Pérez, J. and Saameño-Rodríguez, J.J. (1999) 'Location of an undesirable facility in a polygonal region with forbidden zones', *European Journal of Operational Research*, vol. 114, no. 2, pp. 372-379.

Murray, A.T. and Church, R.L. (1997) 'Solving the anti-covering location problem using Lagrangian relaxation', *Computers & Operations Research*, vol. 24, no. 2, pp. 127-140.

Murty, K.G. and Djang, P.A. (1999) 'The U.S. army national guard's mobile training simulators location and routing problem', *Operations Research*, vol. 47, no. 2, pp. 175-182.

Muyldermans, L. (2003) 'Routing, Districting and Location for Arc Traversal Problems', PhD thesis, Katholieke Universiteit Leuven, Leuven.

Nagy, G. and Salhi, S. (1996a) 'Nested heuristic methods for the location-routeing problem', *Journal of the Operational Research Society*, vol. 47, no. 9, pp. 1166-1174.

Nagy, G. and Salhi, S. (1996b) 'A nested location-routing heuristic using route length estimation', *Studies in Locational Analysis*, vol. 10, pp. 109-127.

Nagy, G. and Salhi, S. (1998) 'The many-to-many location-routing problem', *TOP*, vol. 6, no. 2, pp. 261-275.

Nagy, G. and Salhi, S. (2007) 'Location-routing: issues, models and methods', *European Journal of Operational Research*, vol. 177, no. 2, pp. 649-672.

Nambiar, J.M., Gelders, L.F. and Van Wassenhove, L.N. (1981) 'A large scale location-allocation problem in the natural rubber industry', *European Journal of Operational Research*, vol. 6, no. 2, pp. 183-189.

Nambiar, J.M., Gelders, L.F. and Van Wassenhove, L.N. (1989) 'Plant location and vehicle routing in the Malaysian rubber smallholder sector: a case study', *European Journal of Operational Research*, vol. 38, no. 1, pp. 14-26.

Narula, S.C. and Vassilev, V. (1994) 'An interactive algorithm for solving multiple objective integer linear programming problems', *European Journal of Operational Research*, vol. 79, no. 3, pp. 443-450.

Nema, A.K. and Gupta, S.K. (1999) 'Optimization of regional hazardous waste management systems: an improved formulation', *Waste Management*, vol. 19, no. 7-8, pp. 441-451.

Nielsen, J. (1993) *Usability Engineering*, San Diego: Academic Press.

Nielsen, J. (1994) 'Heuristic evaluation', in Nielsen, J. and Mack, R.L. (ed.) *Usability Inspection Methods*, New York: Wiley.

Ogryczak, W., Studziński, K. and Zorychta, K. (1989) 'A solver for the multi-objective transshipment problem with facility location', *European Journal of Operational Research*, vol. 43, no. 1, pp. 53-64.

Ogryczak, W., Studziński, K. and Zorychta, K. (1992) 'DINAS: a computer-assisted analysis system for multiobjective transshipment problems with facility location', *Computers & Operations Research*, vol. 19, no. 7, pp. 637-647.

Ohsawa, Y. (2000) 'Bicriteria Euclidean location associated with maximin and minimax criteria', *Naval Research Logistics*, vol. 47, no. 7, pp. 581-592.

Ohsawa, Y. and Tamura, K. (2003) 'Efficient location for a semi-obnoxious facility', *Annals of Operations Research*, vol. 123, no. 1-4, pp. 173-188.

Open Geospatial Consortium, Inc. (2010) *Web map service* [Online], Available: http://www.opengeospatial.org/standards/wms [25 Nov 2010].

Or, I. and Pierskalla, W.P. (1979) 'A transportation location-allocation model for regional blood banking', *AIIE Transactions*, vol. 11, no. 2, pp. 86-95.

Palubeckis, G. (2007) 'Iterated tabu search for the maximum diversity problem', *Applied Mathematics and Computation*, vol. 189, no. 1, pp. 371-383.

Parragh, S.N., Doerner, K.F. and Hartl, R.F. (2008a) 'A survey on pickup and delivery problems. Part I: transportation between customers and depot', *Journal für Betriebswirtschaft*, vol. 58, no. 1, pp. 21-51.

Parragh, S.N., Doerner, K.F. and Hartl, R.F. (2008b) 'A survey on pickup and delivery problems. Part II: transportation between pickup and delivery locations', *Journal für Betriebswirtschaft*, vol. 58, no. 2, pp. 81-117.

Pearn, W.-L., Assad, A. and Golden, B.L. (1987) 'Transforming arc routing into node routing problems', *Computers & Operations Research*, vol. 14, no. 4, pp. 285-288.

Perl, J. and Daskin, M.S. (1984) 'A unified warehouse location-routing methodology', *Journal of Business Logistics*, vol. 5, no. 1, pp. 92-111.

Perl, J. and Daskin, M.S. (1985) 'A warehouse location-routing problem', *Transportation Research Part B*, vol. 19, no. 5, pp. 381-396.

Perl, J. and Sirisoponsilp, S. (1988) 'Distribution networks: facility location, transportation and inventory', *International Journal of Physical Distribution & Materials Management*, vol. 18, no. 6, pp. 18-26.

Pia, A.D. and Filippi, C. (2006) 'A variable neighborhood descent algorithm for a real waste collection problem with mobile depots', *International Transactions in Operational Research*, vol. 13, no. 2, pp. 125-141.

Pisinger, D. (2006) 'Upper bounds and exact algorithms for p-dispersion problems', *Computers & Operations Research*, vol. 33, no. 5, pp. 1380-1398.

Plastria, F. (1992) 'GBSSS: the generalized big square small square method for planar single-facility location', *European Journal of Operational Research*, vol. 62, no. 2, pp. 163-174.

Plastria, F. (1996) 'Optimal location of undesirable facilities: a selective overview', *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 36, no. 2-3, pp. 109-127.

Plastria, F. and Carrizosa, E. (1999) 'Undesirable facility location with minimal covering objectives', *European Journal of Operational Research*, vol. 119, no. 1, pp. 158-180.

Polacek, M., Doerner, K.F., Hartl, R.F. and Maniezzo, V. (2008) 'A variable neighborhood search for the capacitated arc routing problem with intermediate facilities', *Journal of Heuristics*, vol. 14, no. 5, pp. 405-423.

Pressman, R.S. (2001) *Software Engineering: A Practitioner's Approach*, 5th edition, New York: McGraw-Hill.

Prins, C., Prodhon, C., Ruiz, A., Soriano, P. and Wolfler Calvo, R. (2007) 'Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation-granular tabu search heuristic', *Transportation Science*, vol. 41, no. 4, pp. 470-483.

Prins, C., Prodhon, C. and Wolfler Calvo, R. (2006) 'Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking', *4OR*, vol. 4, no. 3, pp. 221-238.

Rakas, J., Teodorović, D. and Kim, T. (2004) 'Multi-objective modeling for determining location of undesirable facilities', *Transportation Research Part D*, vol. 9, no. 2, pp. 125-138.

Rand, G.K. (1976) 'Methodological choices in depot location studies', *Operational Research Quarterly*, vol. 27, no. 1, pp. 241-249.

Rasmussen, L.M. (1986) 'Zero—one programming with multiple criteria', *European Journal of Operational Research*, vol. 26, no. 1, pp. 83-95.

Ratick, S.J. and White, A.L. (1988) 'A risk-sharing model for locating noxious facilities', *Environment and Planning B*, vol. 15, no. 2, pp. 165-179.

Resende, M.G.C., Martí, R., Gallego, M. and Duarte, A. (2010) 'GRASP and path relinking for the max–min diversity problem', *Computers & Operations Research*, vol. 37, no. 3, pp. 498-508.

Resende, M.G.C. and Ribeiro, C.C. (2003) 'Greedy randomized adaptive search procedures', in Glover, F.W. and Kochenberger, G.A. (ed.) *Handbook of Metaheuristics*, Norwell: Kluwer Academic Publishers.

ReVelle, C.S., Cohon, J. and Shobrys, D. (1991) 'Simultaneous siting and routing in the disposal of hazardous wastes', *Transportation Science*, vol. 25, no. 2, pp. 138-145.

ReVelle, C.S. and Eiselt, H.A. (2005) 'Location analysis: a synthesis and survey', *European Journal of Operational Research*, vol. 165, no. 1, pp. 1-19.

Riopel, D., Langevin, A. and Campbell, J.F. (2005) 'The network of logistics decisions', in Langevin, A. and Riopel, D. (ed.) *Logistics Systems: Design and Optimization*, New York: Springer.

Romero-Morales, D., Carrizosa, E. and Conde, E. (1997) 'Semi-obnoxious location models: a global optimization approach', *European Journal of Operational Research*, vol. 102, no. 2, pp. 295-301.

Rosenkrantz, D.J., Tayi, G.K. and Ravi, S.S. (2000) 'Facility dispersion problems under capacity and cost constraints', *Journal of Combinatorial Optimization*, vol. 4, no. 1, pp. 7-33.

Rubin, J. and Chisnell, D. (2008) *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*, 2nd edition, Indianapolis: Wiley.

Rushton, A., Croucher, P. and Baker, P. (2006) *The Handbook of Logistics and Distribution Management*, 3rd edition, London: Kogan Page.

Russell, R. (1995) 'Hybrid heuristics for the vehicle routing problem with time windows', *Transportation Science*, vol. 29, no. 2, pp. 156-166.

Saameño-Rodríguez, J.J., García, C.G., Muñoz-Pérez, J. and Casermeiro, E.M. (2006) 'A general model for the undesirable single facility location problem', *Operations Research Letters*, vol. 34, no. 4, pp. 427-436.

Salhi, S. and Fraser, M. (1996) 'An integrated heuristic approach for the combined location vehicle fleet mix problem', *Studies in Locational Analysis*, vol. 8, pp. 3-21.

Salhi, S. and Nagy, G. (1999) 'Consistency and robustness in location-routing', *Studies in Locational Analysis*, vol. 13, pp. 3-19.

Salhi, S. and Nagy, G. (2009) 'Local improvement in planar facility location using vehicle routing', *Annals of Operations Research*, vol. 167, no. 1, pp. 287-296.

Salhi, S. and Rand, G.K. (1989) 'The effect of ignoring routes when locating depots', *European Journal of Operational Research*, vol. 39, no. 2, pp. 150-156.

Savelsbergh, M. (1992) 'The vehicle routing problem with time windows: minimizing route duration', *INFORMS Journal on Computing*, vol. 4, no. 2, pp. 146-154.

Sayin, S. (2009) 'Multi-objective optimization and decision support systems', in Floudas, C.A. and Pardalos, P.M. (ed.) *Encyclopedia of Optimization*, 2nd edition, New York: Springer.

Schittekat, P. and Sörensen, K. (2009) 'Supporting 3PL decisions in the automotive industry by generating diverse solutions to a large-scale location-routing problem', *Operations Research*, vol. 57, no. 5, pp. 1058-1067.

Schwardt, M. and Dethloff, J. (2005) 'Solving a continuous location-routing problem by use of a self-organising map', *International Journal of Physical Distribution & Logistics Management*, vol. 35, no. 6, pp. 390-408.

Schwardt, M. and Fischer, K. (2009) 'Combined location-routing problems ─ a neural network approach', *Annals of Operations Research*, vol. 167, no. 1, pp. 253-269.

Semet, F. (1995) 'A two-phase algorithm for partial accessibility constrained vehicle routing problem', *Annals of Operations Research*, vol. 61, no. 1, pp. 45-65.

Semet, F. and Taillard, E. (1993) 'Solving real-life vehicle routing problems efficiently using tabu search', *Annals of Operations Research*, vol. 41, no. 4, pp. 469-488.

Sharp, H., Rogers, Y. and Preece, J. (2007) *Interaction Design: Beyond Human-Computer Interaction*, 2nd edition, Chichester: Wiley.

Shen, Z.-J.M. and Qi, L. (2007) 'Incorporating inventory and routing costs in strategic location models', *European Journal of Operational Research*, vol. 179, no. 2, pp. 372-389.

Shier, D.R. (1977) 'A min-max theorem for p-center problems on a tree', *Transportation Science*, vol. 11, no. 3, pp. 243-252.

Shin, W.S. and Ravindran, A. (1991) 'Interactive multiple objective optimization: survey I— continuous case', *Computers & Operations Research*, vol. 18, no. 1, pp. 97-114.

Simchi-Levi, D. (1991) 'The capacitated traveling salesman location problem', *Transportation Science*, vol. 25, no. 1, pp. 9-18.

Simchi-Levi, D. and Berman, O. (1987) 'Heuristics and bounds for the travelling salesman location problem on the plane', *Operations Research Letters*, vol. 6, no. 5, pp. 243-248.

Simchi-Levi, D. and Berman, O. (1988) 'A heuristic algorithm for the traveling salesman location problem on networks', *Operations Research*, vol. 36, no. 3, pp. 478-484.

Simchi-Levi, D. and Berman, O. (1990) 'Optimal locations and districts of two traveling salesman on a tree', *Networks*, vol. 20, no. 7, pp. 803-815.

Simon, H.A. (1977) *The New Science of Management Decision*, Upper Saddle River: Prentice Hall.

Singh, N. and Shah, J. (2004) 'Managing tendupatta leaf logistics: an integrated approach', *International Transactions in Operational Research*, vol. 11, no. 6, pp. 683-699.

Skriver, A.J.V. and Andersen, K.A. (2003) 'The bicriterion semi-obnoxious location (BSL) problem solved by an ε-approximation', *European Journal of Operational Research*, vol. 146, no. 3, pp. 517-528.

Soland, R.M. (1979) 'Multicriteria optimization: a general characterization of efficient solutions', *Decision Sciences*, vol. 10, no. 1, pp. 26-38.

Sommerville, I. (2007) *Software Engineering*, 8th edition, Harlow: Addison-Wesley.

Srivastava, R. (1993) 'Alternate solution procedures for the location-routing problem', *Omega*, vol. 21, no. 4, pp. 497-506.

Srivastava, R. and Benton, W.C. (1990) 'The location-routing problem: considerations in physical distribution system design', *Computers & Operations Research*, vol. 17, no. 5, pp. 427-435.

Stowers, C.L. and Palekar, U.S. (1993) 'Location models with routing considerations for a single obnoxious facility', *Transportation Science*, vol. 27, no. 4, pp. 350-362.

Su, C.-T. (1998) 'Locations and vehicle routing designs of physical distribution systems', *Production Planning & Control*, vol. 9, no. 7, pp. 650-659.

Sung, C.S. and Joo, C.M. (1994) 'Locating an obnoxious facility on a Euclidean network to minimize neighborhood damage', *Networks*, vol. 24, no. 1, pp. 1-9.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F. and Potvin, J.-Y. (1997) 'A tabu search heuristic for the vehicle routing problem with soft time windows', *Transportation Science*, vol. 31, no. 2, pp. 170-186.

Talbi, E.-G. (2009) *Metaheuristics: From Design to Implementation*, Hoboken: Wiley.

Tamir, A. (1991) 'Obnoxious facility location on graphs', *SIAM Journal on Discrete Mathematics*, vol. 4, no. 4, pp. 550-567.

Tamir, A. (2006) 'Locating two obnoxious facilities using the weighted maximin criterion', *Operations Research Letters*, vol. 34, no. 1, pp. 97-105.

Teghem, J. (2009) 'Multi-objective integer linear programming', in Floudas, C.A. and Pardalos, P.M. (ed.) *Encyclopedia of Optimization*, 2nd edition, New York: Springer.

Teixeira, L. (2008) 'Contribuições para o desenvolvimento de sistemas de informação na saúde: aplicação na área da hemofilia', PhD thesis, University of Aveiro, Aveiro [in Portuguese].

Tompkins, J.A., White, J.A., Bozer, Y.A. and Tanchoco, J.M.A. (2010) *Facilities Planning*, 4th edition, Hoboken: Wiley.

Toregas, C., Swain, R., ReVelle, C.S. and Bergman, L. (1971) 'The location of emergency service facilities', *Operations Research*, vol. 19, no. 6, pp. 1363-1373.

Toth, P. and Vigo, D. (ed.) (2002) *The Vehicle Routing Problem*, Philadelphia: Society for Industrial and Applied Mathematics.

Tsang, E. and Voudouris, C. (1997) 'Fast local search and guided local search and their application to British telecom's workforce scheduling problem', *Operations Research Letters*, vol. 20, no. 3, pp. 119-127.

Turban, E., Aronson, J.E., Liang, T.-P. and Sharda, R. (2007) *Decision Support and Business Intelligence Systems*, 8th edition, Upper Saddle River: Pearson-Prentice Hall.

Tuzun, D. and Burke, L.I. (1999) 'A two-phase tabu search approach to the location routing problem', *European Journal of Operational Research*, vol. 116, no. 1, pp. 87-99.

Vassilev, V. and Narula, S.C. (1993) 'A reference direction algorithm for solving multiple objective integer linear programming problems', *Journal of the Operational Research Society*, vol. 44, no. 12, pp. 1201-1209.

Voudouris, C. (1997) 'Guided Local Search for Combinatorial Optimisation Problems', PhD thesis, University of Essex, Colchester.

Voudouris, C. and Tsang, E. (1999) 'Guided local search and its application to the traveling salesman problem', *European Journal of Operational Research*, vol. 113, no. 2, pp. 469-499.

Ware, C. (2004) *Information Visualization: Perception for Design*, 2nd edition, San Francisco: Morgan Kaufmann Publishers.

Wasner, M. and Zäpfel, G. (2004) 'An integrated multi-depot hub-location vehicle routing model for network planning of parcel service', *International Journal of Production Economics*, vol. 90, no. 3, pp. 403-419.

Watson-Gandy, C.D.T. and Dohrn, P.J. (1973) 'Depot location with van salesmen — a practical approach', *Omega*, vol. 1, no. 3, pp. 321-329.

Wells, D. (2009) *Extreme programming: a gentle introduction* [Online], Available: http://www.extremeprogramming.org/ [25 Nov 2010].

Whitten, J.L. and Bentley, L.D. (2007) *Systems Analysis and Design Methods*, 7th edition, New York: McGraw-Hill/Irwin.

Wiegers, K.E. (2003) *Software Requirements*, 2nd edition, Redmond: Microsoft Press.

Wøhlk, S. (2008) 'A decade of capacitated arc routing', in Golden, B.L., Raghavan, S. and Wasil, E. (ed.) *The Vehicle Routing Problem: Latest Advances and New Challenges*, New York: Springer.

Wood, W.A. and Kleb, W.L. (2002) 'Extreme programming in a research environment', in Wells, D. and Williams, L. (ed.) *Extreme Programming and Agile Methods — XP/Agile Universe 2002*, Berlin: Springer-Verlag.

World Wide Web Consortium (2010) *XML technology - W3C* [Online], Available: http://www.w3.org/standards/xml/ [25 Nov 2010].

Wu, T.-H., Low, C. and Bai, J.-W. (2002) 'Heuristic solutions to multi-depot location-routing problems', *Computers & Operations Research*, vol. 29, no. 10, pp. 1393-1415.

Wyman, M.M. and Kuby, M. (1995) 'Proactive optimization of toxic waste transportation, location and technology', *Location Science*, vol. 3, no. 3, pp. 167-185.

Yapicioglu, H., Smith, A.E. and Dozier, G. (2007) 'Solving the semi-desirable facility location problem using bi-objective particle swarm', *European Journal of Operational Research*, vol. 177, no. 2, pp. 733-749.

Yu, V.F., Lin, S.-W., Lee, W. and Ting, C.-J. (2010) 'A simulated annealing heuristic for the capacitated location routing problem', *Computers & Industrial Engineering*, vol. 58, no. 2, pp. 288-299.

Zhang, F.G. and Melachrinoudis, E. (2001) 'The maximin-maxisum network location problem', *Computational Optimization and Applications*, vol. 19, no. 2, pp. 209-234.

Zmazek, B. and Žerovnik, J. (2004) 'The obnoxious center problem on weighted cactus graphs', *Discrete Applied Mathematics*, vol. 136, no. 2-3, pp. 377-386.

Zografos, K.G. and Samara, S. (1989) 'A combined location-routing model for hazardous waste transportation and disposal', *Transportation Research Record*, vol. 1245, pp. 52-59.

Appendices

A. A Brief Analysis of the Location-Routing Problem Literature

This analysis aims to help the reader understand the research trends in the location-routing problem (LRP) and identify the core journals for the subject. Firstly, the number of publications per journal is analysed. Then, some information on the evolution of the research over the years is presented. Finally, this information is cross-analyzed in order to present the publication evolution and identify the most significant journals for future publications.

Some journals have changed their titles over the years. This was the case of "Operational Research Quarterly", called "Journal of the Operational Research Society" since 1978, "AIIE Transactions" currently named "IIE transactions", "Naval Research Logistics Quarterly" that became "Naval Research Logistics", and finally "4OR A Quarterly Journal of Operations Research", which appeared in 2003 replacing "JORBEL" and "Ricerca Operativa", the journals of the Belgian and Italian operations research societies respectively. These changes were taken into account in this analysis by considering both the older and the more recent versions of the journals as a single publication (since invariably the newer publications led to the discontinuation of the older ones). Table A.1 lists for each journal the total number of papers devoted to LRP and the number of papers per year.

Analysing Figure A.1 it is possible to observe the evolution of LRP publications over the years. It can be concluded that ever since the 1970s there has been a constant increase of works in this area. Another observation that can be made is regarding the algorithmic approach: over the years there appears to be a shift of focus from the exact studies to the heuristic approaches (probably due to the reasons mentioned in Section 2.2.3).

Finally, this analysis focuses on the journals with the most publications in the area. Looking at Table A.1 it can be concluded that around fifty percent of the publications can be found in only four journals: "European Journal of Operational Research" (EJOR), "Transportation Science" (TS), "Computers & Operations Research" (C&OR), and "Journal of the Operational Research Society" (JORS), formerly named "Operational Research Quarterly".

Figure A.2 shows the evolution of LRP publications in the four core journals for the subject. One can observe the pioneer and somewhat steady (although not very significant) stream of publications in JORS. It is also possible to observe the "competition" between EJOR and TS (from 1970 to 2000). In the last decade however, TS has lost some of its relevance in LRPs to C&OR.

Table A.1 Publications per journal (and year of publication).

Journal	Total	Year[number of articles]
European Journal of Operational Research	25	1980, 1981[2], 1983[2], 1988[2], 1989[3], 1990, 1994, 1995, 1996[2], 1998, 1999, 2005[2], 2006, 2007[4], 2008
Transportation Science	18	1976[2], 1977, 1982, 1985[2], 1988[2], 1989[2], 1991[3], 1993, 1994, 1995, 2007[2]
Computers & Operations Research	11	1990, 1992, 2001, 2002[2], 2005[2], 2007, 2008, 2009, 2010
Operational Research Quarterly / Journal of the Operational Research Society	9	1964, 1969, 1989, 1992, 1994, 1996, 2004, 2006, 2008
Annals of Operations Research	7	1986, 1993, 1995, 2002, 2009[3]
Networks	5	1986, 1988, 1990, 1995, 1999
Operations Research	5	1972, 1988, 1990, 1999, 2009
Studies in Locational Analysis	4	1993, 1996[2], 1999
AIIE Transactions / IEE Transactions	2	1979, 1982
Belgian Journal of Operational Research, Statistics and Computer Science / 4OR	2	1986, 2006
Computers & Industrial Engineering	2	2002, 2010
Computers & Mathematics with Applications	2	1977, 1978
International Journal of Advanced Manufacturing Technology	2	2003, 2005
Journal of Business Logistics	2	1984, 1996
Location Science	2	1995[2]
Naval Research Logistics Quarterly / Naval Research Logistics	2	1985, 1994
Omega	2	1973, 1993
Asia-Pacific Journal of Operational Research	1	2001
Decision Sciences	1	1993
Discrete Applied Mathematics	1	2004
INFOR	1	1989
Interfaces	1	1999
International Journal of Industrial Engineering	1	2002
International Journal of Logistics Research and Applications	1	2008
International Journal of Physical Distribution & Logistics Management	1	2005
International Journal of Production Economics	1	2004
International Journal of Systems Applications, Engineering & Development	1	2007
International Transactions in Operational Research	1	2004
Journal of Heuristics	1	2005
Journal of Mathematical Modelling and Algorithms	1	2008
Journal of Modelling Management	1	2006
Journal of Regional Science	1	1976
Journal of Urban Planning and Development	1	1994
Mathematical and Computer Modelling	1	2005
OPSEARCH	1	2001
Operations Research Letters	1	1987
OR Spektrum	1	2000
Photonic Network Communications	1	2003
Production Planning & Control	1	1998
TOP	1	1998
Transportation Research Part B	1	1985
Transportation Research Record	1	1989
Waste Management	1	1999

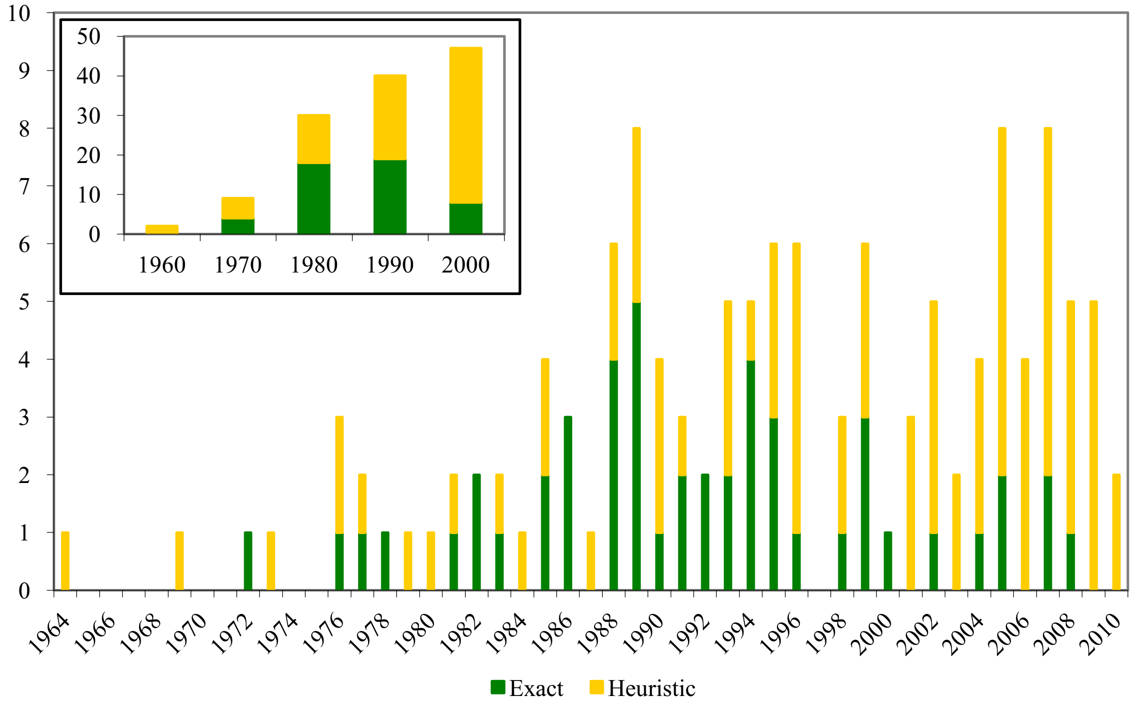


Figure A.1 LRP publications by year and decade, categorized by algorithmic approach (exact and heuristic).

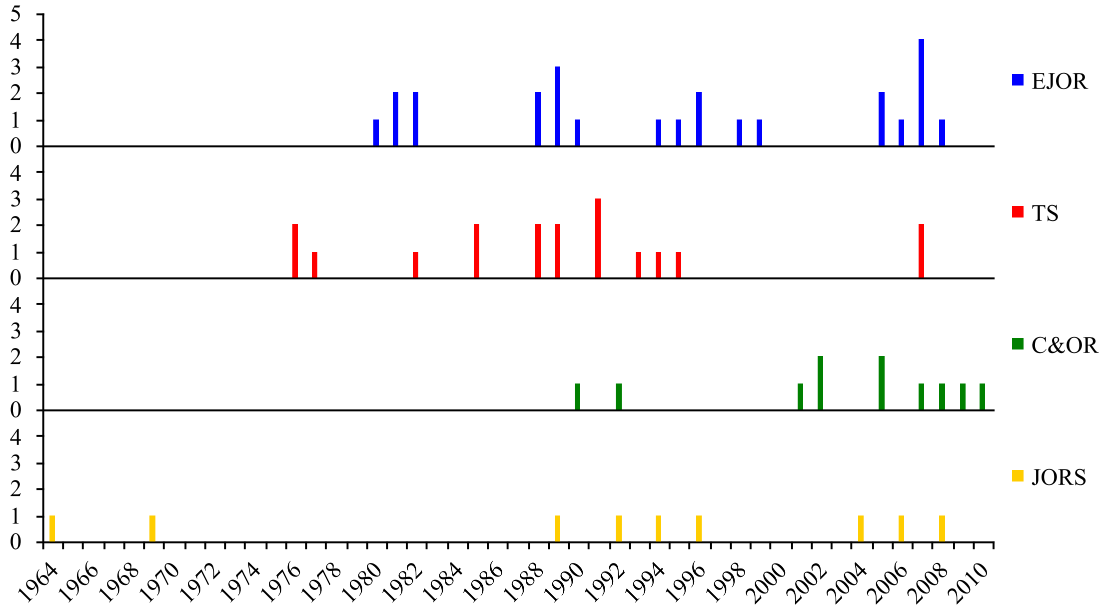


Figure A.2 LRP publications by year in the core journals.

B. Test Instance Data

In this appendix data regarding the test instance (used in the step-by-step example of Section 5.3.2) are presented. The data were generated having in mind the characteristics of a multi-objective capacitated LRP.

Following the notation given in Sections 3.1.1 and 4.3.1, the test instance is composed of 10 clients (n) and the number m of depots is 5. Data regarding the clients can be seen in Table B.1. The first three columns display, for each client, respectively, the number, x and y coordinates. For each client (community) $i \in I$ there is a demand d_i and a size s_i, which can be seen in the last two columns. The depots' data are shown in Table B.2 where, similarly to Table B.1, the first three columns characterize the depot's number, x and y coordinate. In the last two columns of Table B.2 it can be seen the capacity w_j and the cost f_j, for each depot $j \in J$. Finally, the vehicle capacity $Q = 140$ and there is no vehicle fixed cost ($F = 0$).

Table B.1 Data regarding the clients of the test instance.

Client	X	Y	d_i	s_i
1	24	33	60	60
2	29	32	40	40
3	17	29	20	20
4	9	28	40	40
5	22	26	20	20
6	31	25	20	20
7	30	17	20	20
8	15	16	60	60
9	15	9	40	40
10	27	9	20	20

Table B.2 Data regarding the depots of the test instance.

Depot	X	Y	w_j	f_j
1	16	31	280	4
2	26	29	280	5
3	16	21	280	10
4	28	21	280	24
5	21	12	280	20

C. Non-Dominated and Efficient Solutions of the Test Instance

All the non-dominated solutions of the test instance (used in the step-by-step example of Section 5.3.2) can be seen in this appendix, where a total of 9 exists. The data of the test instance are available in Appendix B.

Table C.1 displays, for each of the non-dominated solutions, the objective function values (Z^i, $i = 1,\ 2,\ 3$). The graphical representation corresponding to each of the efficient solutions is depicted in Figure C.1.

Table C.1 Non-dominated solutions of the test instance.

Solution	Z^1	Z^2	Z^3
S^1	115.0240	18.3037	0.227730
S^2	155.5355	6.8483	0.059434
S^3	138.4380	7.9179	0.054717
S^4	143.0114	7.5433	0.057692
S^5	117.0605	11.6599	0.059004
S^6	115.0433	13.9973	0.212346
S^7	116.6083	10.9503	0.215385
S^8	123.3988	9.8807	0.203279
S^9	125.6691	10.5903	0.057711

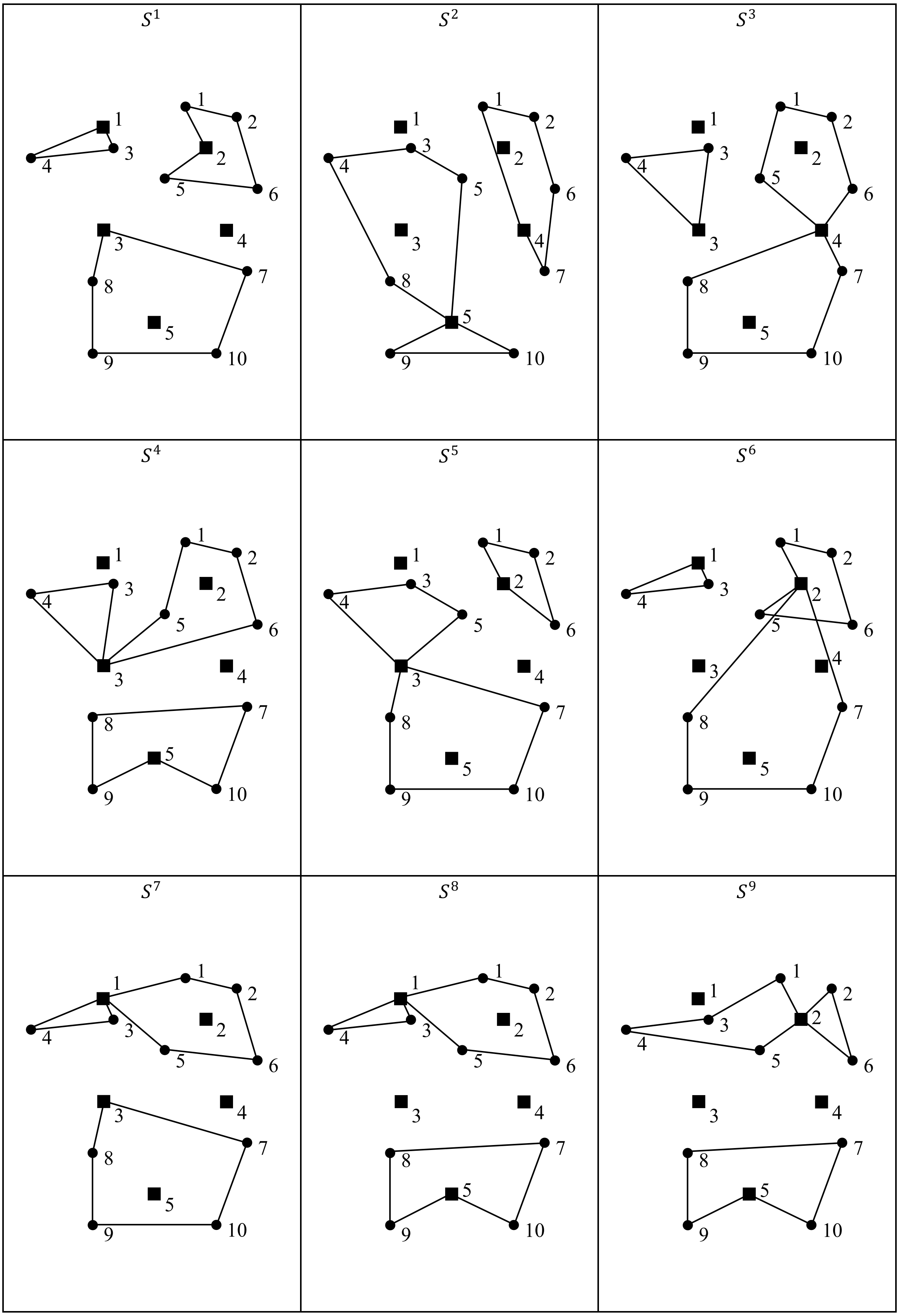


Figure C.1 Graphical representation of the efficient solutions of the test instance.

D. Decision-Support Tool Data Structure

In the following figures it can be seen the data structure, adopted for the decision-support tool, for the depots (Figure D.1), vehicles (Figure D.2), graphs (Figure D.3), and solutions (Figure D.4).

```
  <?xml version="1.0" ?>
  <!--  Created by Rui Borges Lopes (c) University of Aveiro  -->
- <Depots>
  - <Depot>
      <Number>1</Number>
      <Name>Leiria</Name>
      <XCoord>1320</XCoord>
      <YCoord>3224</YCoord>
      <Capacity>12000</Capacity>
      <FixedCost>100</FixedCost>
      <VariableCost>0</VariableCost>
      <Installed>True</Installed>
      <Level>1</Level>
      <Colour>#C0FF0000</Colour>
    </Depot>
  - <Depot>
      <Number>2</Number>
      <Name>Entroncamento</Name>
      <XCoord>1650</XCoord>
      <YCoord>3073</YCoord>
      <Capacity>12000</Capacity>
      <FixedCost>150</FixedCost>
      <VariableCost>0</VariableCost>
      <Installed>False</Installed>
      <Level>2</Level>
      <Colour>#C080C0C0</Colour>
    </Depot>
  </Depots>
  <!--  Depots:        Depots                                              -->
  <!--  Depot:         Depot                                               -->
  <!--  Number:        Depot number                          int           -->
  <!--  Name:          Depot name                            string        -->
  <!--  XCoord:        Depot x coordinate                    double        -->
  <!--  YCoord:        Depot y coordinate                    double        -->
  <!--  Capacity:      Depot capacity                        double        -->
  <!--  FixedCost:     Depot fixed cost                      double        -->
  <!--  VariableCost:  Depot variable cost                   double        -->
  <!--  Installed:     Depot has already been installed      boolean       -->
  <!--  Level:         Depot level (primary, secondary, etc.)  unsignedByte  -->
  <!--  Colour:        Depot colour                          string        -->
```

Figure D.1 Data structure of the depots file.

```
  <?xml version="1.0" ?>
  <!--  Created by Rui Borges Lopes (c) University of Aveiro  -->
- <Vehicles>
  - <Vehicle>
      <Number>1</Number>
      <Name>Ford 12-AA-25</Name>
      <Capacity>1000</Capacity>
      <Availability>5</Availability>
      <FixedCost>15</FixedCost>
      <VariableCost>1</VariableCost>
      <DirectTour>False</DirectTour>
      <RteLength>0</RteLength>
      <RteDuration>0</RteDuration>
      <Level>1</Level>
    </Vehicle>
  - <Vehicle>
      <Number>2</Number>
      <Name>Opel 84-74-PG</Name>
      <Capacity>750</Capacity>
      <Availability>12</Availability>
      <FixedCost>9</FixedCost>
      <VariableCost>0</VariableCost>
      <DirectTour>True</DirectTour>
      <RteLength>300</RteLength>
      <RteDuration>0</RteDuration>
      <Level>1</Level>
    </Vehicle>
  </Vehicles>
  <!--  Vehicles:      Vehicles                                                  -->
  <!--  Vehicle:       Vehicle                                                   -->
  <!--  Number:        Vehicle number                          int               -->
  <!--  Name:          Vehicle name                            string            -->
  <!--  Capacity:      Vehicle capacity                        double            -->
  <!--  Availability:  Number of available similar vehicles    double            -->
  <!--  FixedCost:     Fixed cost for each used vehicle        double            -->
  <!--  VariableCost:  Vehicle variable cost                   double            -->
  <!--  DirectTour:    Vehicle can only perform a direct tour  boolean           -->
  <!--  RteLength:     Vehicle maximum route length allowed    double            -->
  <!--  RteDuration:   Vehicle maximum route duration allowed  double            -->
  <!--  Level:         Level in which the vehicle operates     unsignedByte      -->
```


Figure D.2 Data structure of the vehicles file.

```
  <?xml version="1.0" ?>
  <!--  Created by Rui Borges Lopes (c) University of Aveiro  -->
- <Graphs>
  - <Graph>
      <Number>1</Number>
      <Name>Distance matrix</Name>
      <Objective>Cost</Objective>
      <Complete>True</Complete>
      <Directed>False</Directed>
    - <Arcs>
        <SrcObject>C</SrcObject>
        <SrcNumber>2</SrcNumber>
      - <Des>
          <Obj>C</Obj>
          <Num>1</Num>
          <Len>55.8</Len>
        </Des>
      - <Des>
          <Obj>D</Obj>
          <Num>3</Num>
          <Len>250</Len>
        </Des>
      </Arcs>
    - <Arcs>
        <SrcObject>D</SrcObject>
        <SrcNumber>3</SrcNumber>
      - <Des>
          <Obj>C</Obj>
          <Num>2</Num>
          <Len>250</Len>
        </Des>
      </Arcs>
    </Graph>
  - <Graph>
      <Number>2</Number>
      <Name>Obnoxious effect matrix</Name>
      <Objective>Obnoxious effect</Objective>
      <Complete>True</Complete>
      <Directed>True</Directed>
    - <Arcs>
        <SrcObject>D</SrcObject>
        <SrcNumber>1</SrcNumber>
      - <Des>
          <Obj>C</Obj>
          <Num>4</Num>
          <Len>13.45</Len>
        </Des>
      </Arcs>
    </Graph>
  </Graphs>
  <!--  Graphs:       Graphs                                         -->
  <!--  Graph:        Graph                                          -->
  <!--  Number:       Graph number                      unsignedByte -->
  <!--  Name:         Graph name                        string       -->
  <!--  Objective:    Corresponding objective name      string       -->
  <!--  Complete:     Graph is complete                 boolean      -->
  <!--  Directed:     Graph is directed                 boolean      -->
  <!--  Arcs:         Arcs with the same source node                 -->
  <!--  SrcObject:    Arcs source node object type      string       -->
  <!--                C = Client | D = Depot                         -->
  <!--  SrcNumber:    Arcs source node object number    int          -->
  <!--  Des:          Arc destination node                           -->
  <!--  Obj:          Arc destination node object type  string       -->
  <!--                C = Client | D = Depot                         -->
  <!--  Num:          Arc destination node object number int         -->
  <!--  Len:          Source node to destination length double       -->
```


Figure D.3 Data structure of the graphs file.

```
  <?xml version="1.0" ?>
  <!--  Created by Rui Borges Lopes (c) University of Aveiro  -->
- <Solutions>
  - <Solution>
      <Number>1</Number>
      <Name>CLRP hybrid extended savings algorithm</Name>
      <Time>P0DT1H27M14.3998722S</Time>
    - <Objective>
        <Number>1</Number>
        <Name>Cost</Name>
        <Value>15.445</Value>
      </Objective>
    - <Route>
        <Number>1</Number>
        <Name>Train</Name>
        <Capacity>970</Capacity>
        <VhcNumber>2</VhcNumber>
        <Colour>#0080C0C0</Colour>
      - <Objective>
          <Number>1</Number>
          <Name>Cost</Name>
          <Value>7.8</Value>
        </Objective>
      - <Stop>
          <Order>0</Order>
          <Object>D</Object>
          <ObjNumber>1</ObjNumber>
          <Service>P</Service>
          <SrvAmount>970</SrvAmount>
          <Fixed>False</Fixed>
        </Stop>
      - <Stop>
          <Order>1</Order>
          <Object>C</Object>
          <ObjNumber>4</ObjNumber>
          <Service>D</Service>
          <SrvAmount>970</SrvAmount>
          <Fixed>False</Fixed>
        </Stop>
      </Route>
    </Solution>
  </Solutions>
  <!--  Solutions:    Solutions                                          -->
  <!--  Solution:     Solution                                           -->
  <!--  Number:       Solution number                        int         -->
  <!--  Name:         Solution name                          string      -->
  <!--  Time:         Time needed to obtain the solution     duration    -->
  <!--  Objective:    Objective                                          -->
  <!--  Number:       Objective number                       unsignedByte -->
  <!--  Name:         Objective name                         string      -->
  <!--  Value:        Objective value                        double      -->
  <!--  Route:        Solution route                                     -->
  <!--  Number:       Solution route number                  int         -->
  <!--  Name:         Solution route name                    string      -->
  <!--  Capacity:     Solution route total used capacity     double      -->
  <!--  VhcNumber:    Number of the assigned vehicle         int         -->
  <!--  Colour:       Solution route colour                  string      -->
  <!--  Stop:         Solution route stop                                -->
  <!--  Order:        Stopping order (0 stands for departure) int        -->
  <!--  Object:       Object type in which the stop was made string      -->
  <!--                C = Client | D = Depot                             -->
  <!--  ObjNumber:    Route stop object number               int         -->
  <!--  Service:      Service type performed in the stop     string      -->
  <!--                D = Delivery | P = Pickup                          -->
  <!--  SrvAmount:    The amount serviced in the stop        double      -->
  <!--  Fixed:        The corresponding stop order is fixed  boolean     -->
```


Figure D.4 Data structure of the solutions file.